Fall 2023

Application of Big Data Technology, Text Classification, and Azure Machine Learning for Financial Risk Management Using Data Science Methodology

Oluwaseyi A. Ijogun

Recommended Citation

Ijogun, Oluwaseyi A., "Application of Big Data Technology, Text Classification, and Azure Machine Learning for Financial Risk Management Using Data Science Methodology" (2023). *Electronic Theses and Dissertations*. 2654.
https://digitalcommons.georgiasouthern.edu/etd/2654

APPLICATION OF BIG DATA TECHNOLOGY, TEXT CLASSIFICATION, AND AZURE MACHINE LEARNING FOR FINANCIAL RISK MANAGEMENT USING DATA SCIENCE METHODOLOGY

by

OLUWASEYI ADEMOLA IJOGUN

(Under the Direction of Hayden Wimmer)

ABSTRACT

Data science plays a crucial role in enabling organizations to optimize data-driven opportunities within financial risk management. It involves identifying, assessing, and mitigating risks, ultimately safeguarding investments, reducing uncertainty, ensuring regulatory compliance, enhancing decision-making, and fostering long-term sustainability. This thesis explores three facets of Data Science projects: enhancing customer understanding, fraud prevention, and predictive analysis, with the goal of improving existing tools and enabling more informed decision-making. The first project examined leveraged big data technologies, such as Hadoop and Spark, to enhance financial risk management by accurately predicting loan defaulters and their repayment likelihood. In the second project, we investigated risk assessment and fraud prevention within the financial sector, where Natural Language Processing and machine learning techniques were applied to classify emails into categories like spam, ham, and phishing. After training various models, their performance was rigorously evaluated. In the third project, we explored the utilization of Azure machine learning to identify loan defaulters, emphasizing the comparison of different machine learning algorithms for predictive analysis. The results aimed to determine the best-performing model by evaluating various performance metrics for the dataset. This study is important because it offers a strategy for enhancing risk management, preventing fraud, and encouraging innovation in the financial industry, ultimately resulting in better financial outcomes and enhanced customer protection.

INDEX WORDS: Big data technology, Text analysis, Data science, Cyber security, Machine learning, NLP, Azure machine learning, Financial risk management, Hadoop, Spark.

APPLICATION OF BIG DATA TECHNOLOGY, TEXT CLASSIFICATION, AND AZURE MACHINE LEARNING FOR FINANCIAL RISK MANAGEMENT USING DATA SCIENCE METHODOLOGY

by

OLUWASEYI ADEMOLA IJOGUN

M.S., Georgia Southern University, 2023

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

APPLICATION OF BIG DATA TECHNOLOGY, TEXT CLASSIFICATION, AND AZURE MACHINE LEARNING FOR FINANCIAL RISK MANAGEMENT USING DATA SCIENCE METHODOLOGY

by

OLUWASEYI ADEMOLA IJOGUN

Electronic Version Approved:

December 2023

DEDICATION

To God, who has been seeing me through my life pursuit. I also love to dedicate this research to my late father Mr. Oluwafemi Ijogun whose continued legacy and watchwords for excellence have kept me going.

ACKNOWLEDGMENTS

I have had the privilege of working with and learning from a remarkable group of accomplished individuals who provided unwavering moral and academic support, contributing significantly to the successful completion of my thesis research. I extend my heartfelt gratitude to my mother Mrs. Olabisi Ijogun, my Siblings, esteemed professor Hayden Wimmer, dedicated committee members, friends, and family for their continued support and encouragement throughout this research journey. I also wish to express my profound appreciation to Mr. Kikiope Oyewole for his instrumental role in motivating and supporting me throughout my entire degree program. His guidance and encouragement have been invaluable in shaping my academic pursuits.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1

1 INTRODUCTION

Advanced technology has streamlined financial operations, allowing digital processes with minimal human involvement. Financial institutions utilize these advancements to provide online lending and financial services. However, this digital shift brings new risks, surpassing the effectiveness of traditional risk management. Given the intricacies of financial risk management, institutions persistently strive to identify and mitigate risks. Financial risk management is a crucial aspect of safeguarding an organization's financial stability and preventing potential losses. Effective financial management not only leads to cost savings but also enhances decision-making and ultimately results in improved returns. Proactive measures enable companies to stay ahead of potential threats. The surge in loan applications for banks and financial institutions has led to an increase in bad credit losses, posing a risk to creditors' capital and the institutions' reputation and sustainability. Loans are sought for various purposes, including consumer needs, education, medical expenses, travel, and business ventures. Banks and financial institutions can gain valuable insights into customers' spending habits, financial behavior, default prediction, and other characteristics through different modeling techniques. While numerous studies have attempted to predict the probability of loan defaults, these mechanisms have proven insufficient as default rates continue to rise. This research addresses the critical factors influencing loan prediction and highlights the significant impact of effective credit risk management on banking profitability. It employs tested and well-suited machine learning algorithms to achieve this. Additionally, it explores how financial institutions can prevent fraud by effectively classifying emails, thereby mitigating the threats posed by spam and phishing emails.

In the first research we leveraged big data technologies, such as Hadoop and Spark, to enhance financial risk management by accurately predicting loan defaulters and their repayment likelihood. The objective of this research is to effectively manage credit risk in finance using a big data approach, where customer’s spending habits and profiles are being computationally analyzed in other to accurately predict the tendency of defaulting in credit facilities. The methods used in this research have been proven to accurately predict and classify bank customers into defaulters and non-defaulters by using machine learning to train our data set. The method used in this research utilized the efficiency of HDFS in storage of the dataset on a virtual machine using Apache spark as the processing engine for processing big data. The data underwent various preprocessing steps, which included data cleaning and the removal of duplicates. To prepare the dataset for machine learning algorithms, it was further divided into training and testing sets using a train-test split approach. This division facilitates effective classification and model evaluation. To determine how good the model is, different machine learning algorithms were used to determine the most effective and efficient model such as the Decision tree, Naïve Bayes, and Random Forest technique. The accuracy, precision, recall, F1 score, and the confusion matrices were also used to determine the performance for which the true values were known. The result is evaluated for each classifier and the best performing algorithm was determined. Radom Forest gave the best output among other machine learning algorithm tested. The result suggested the machine learning algorithm with the highest Recall, accuracy, and precision as the most efficient model in predicting the effectiveness of the model to predict the tendency of default of bank customers.

In the second research we investigated risk assessment and fraud prevention within the financial sector, where Natural Language Processing and machine learning techniques were applied to classify emails into categories like spam, ham, and phishing. A lot of research has been done on developing different techniques for email classification using machine learning on a spam-ham dataset, however the novelty of this research considers introducing a spam-ham-phishing dataset using machine learning and NLP techniques on text data to improve email classification. To achieve the objective of the study a spam-ham, spam-phishing and ham-phishing dataset was developed using various public data sources, then the dataset was preprocessed, and the key features were selected for machine learning training and testing. Due to the imbalanced dataset used where the ham dataset significantly outweighs the number of spam and phishing dataset, a comparison of the effectiveness of SMOTE and NON-SMOTE technique was also analyzed. Five different machine learning algorithms (SVM, XGBoost, Random Forest, Multinomial and Gaussian Naïve bayes) were used to compare its efficiency using different performance indexes such as accuracy, precision, f1-score, and recall. The result of the model shows that SMOTE gives a better classification performance than the NON-SMOTE technique. However, among the different machine learning algorithms used on the three different datasets, the XGBoost outperforms the other algorithms for the dataset considered.

In the third project, we explored the application of Azure machine learning for the identification of loan defaulters, with a particular focus on comparing various machine learning algorithms for predictive analysis. This exploration took place within the Azure machine learning studio, making use of a range of designer components provided by Azure to enhance the analysis. The project entailed the utilization of diverse machine learning algorithms, including two-class decision forest, two-class neural networks, and two-class support vector machine, all employed in a supervised learning context to ascertain binary classification within the credit dataset. To determine the performance of these machine learning algorithms, a comprehensive evaluation was conducted using five distinct metrics. These metrics encompassed accuracy, precision, recall, F1-score, and the analysis of the confusion matrix. This project aimed to determine which algorithm yielded the most effective results for identifying loan defaulters, providing valuable insights into the power and limitations of various machine learning approaches in the realm of credit risk management. The results clearly demonstrate that the Two-class Decision Forest algorithm outperformed its counterparts, exhibiting exceptional capabilities in generalizing the dataset.

In conclusion, this research provides a crucial approach for strengthening risk management, preventing fraud, and encouraging financial innovation. It suggests strategies to completely transform risk management processes, ensuring improved financial outcomes and customer safety. It also illustrates directions toward an innovative financial environment that is more secure and effective.

CHAPTER 2

2 LITERATURE REVIEW

2.1 Study 1 – Effective Credit Risk Management in Finance Using Big Data Technology

Zhao (2021) studied Big Data Financial Algorithm Technology Based on Machine Learning Technology. Big data has been used to solve problems in the financial sector such as financial instability, business fraud, and poor management. This has directly affected investors because they have to bear huge financial losses. However, there have been limitations in carrying out detailed research due to high research costs & scarcity of data. The objective of this research is to use machine learning techniques to explore and research big data financial algorithms, analyze risk control, and measure the improvement and perfection of traditional financial analysis. The method used for data training was MACD – moving average convergence divergence. The data of user's behavior are respectively collected and summarized using the column expansion method and used into a piece of training data. The result shows that machine learning technology combined with big data for financial algorithms has a strong interpretation ability for the analysis of user behavior (Zhao, 2021).

Peng (2019) studied Stock Analysis And Prediction Using Big Data Analytics, He expressed the role of Big data and its impact on banking and the financial sector to track financial activities, high-frequency trading, catching illegal trading, and curb money laundering and other financial fraud. The objective of the study was to analyze stocks to forecast daily profits in the stock market-based. He showed how big data can be used in predicting stock analysis by using different machine learning tools to train the data sets. He used a Cloudera-Hadoop-based data pipeline approach through data characterization, data injection (flume), storage (HDFS on Cloudera), and pre-processing (pyspark) to perform the analysis for any type and scale of data via real-time data, process it to produce valuable information to support decision making. Using 13 stocks in the us oil fund, the dataset was divided into training and test data to predict the stocks with high daily gains using the machine learning module of spark. Linear regression was done on the dataset then the correlation between stock prices was gotten based on coefficients in the regression model. The R-squared and mean average error were determined. The result showed that there was a positive correlation between stock prices based on the regression model. It identifies stocks with positive everyday return margins (Peng, 2019).

Serengil et al. (2021) studied the machine learning approaches for nonperforming loan prediction. he defined a non-performing loan to be a delay in repayment of a loan within a schedule and define the period of time. If the loan is nonperforming the probability of the borrower repaying the loan is low, this has an extremely negative effect on the sustainability of the banks. The objective of the study is to predict if a customer loan will be payable in a healthy way or not, for the period reviewed. The method used ranges from training based on traditional machine learning and an Imbalance Accuracy Metric (IAM) was also used to gain insight into the predictors. The data set used was from a private bank in Turkey, with emphasis on customer's payment behavior, history, balance sheets, and previous credit and card payments data used for the modeling. The data was later cleansed to remove irrelevant data. Machine learning algorithms such as logistic regression, random forest, support vector machines, bagging classifier, gradient boosting (XGBoost tree algorithm), light GBM tree algorithm, LSTM were performed. Fig 1.0 shows a summary of the different models used and their performance metrics.

Models	Performance Metrics
Logistic Regression	AUC
Random Forest Classifiers	Precision
Support Vector Machines	Recall
Bagging Classifier	F1 Score
LGBN	Imbalance Accuracy
XGBoost	
LSTM	

Table 1 Summary of model methods and performance metrics (Armel & Zaidouni, 2019)

The result using the above model shows samples consisting of 181,567 and 705 features, and the best machine learning algorithm for the time series was determined. The result showed that the lightGBM was the most preferred method among them (Serengil et al., 2021).

Yadav and Thakur (2017) proposed Bank Loan Analysis Using Customer Usage Data: A Big Data Approach Using Hadoop. In his paper, he explained how important the role of the bank is to households, companies, and the economy through the provision of loans to aid and boost their demanding financial needs. Due to the riskiness of lending loans to customers, thorough attention needs to be paid to the disbursement of loans to avoid bad debt or nonperforming loans which directly affect the reliability of the banks. The goal of the lender is to get a return on the investment of his capital. The objective of the research was to find the credit riskiness of the customer usage data by analyzing the data set of a lending club (one of the world's largest online banking marketplaces). To analyze the credit risk related to the customer the big data approach was used to analyze data from different parameters. Data was loaded to a virtual machine called Cloudera software and the same data was injected into the HDFS where it was stored for further analysis. The hive data warehouse tool was used to manage and refine data. An analytics model was used to distinguish and analyze the data, which shows the defaulters to be the least in number. The result showed that there is a good opportunity for the borrowers to get a loan and the investors to get higher interest with lesser risk (Yadav & Thakur, 2017).

Hindistan et al. (2019) studied credit scoring and classification using machine learning techniques on big data platforms. The author discussed the importance of evaluating and analyzing credit risk to investors before issuing any loan. He stressed the credit scoring method as a way to reduce potential risk in financial management. The objective of the paper is to analyze the machine learning and credit scoring methods on big data platforms, he noted that the higher the credit score the lower the risk, and the lower the credit score the higher risk. He simply based it on the customer's integrity and capacity to repay back the loan taken. This study focuses on using big data methodology to analyze the credit score of customers using machine learning techniques. He categorized the classes of loans to be good loans and Bad loans and subclassified them into (fully paid, current, and does not meet credit policy status). A machine learning algorithm was used to analyze by dividing the test and train data to be 20% to 80% respectively. Methods such as Logistic Regression, Random Forest and Decision Tree on HDFS were used to test the data sets. The result shows that each result could be used to analyze the credit risk, however, Logistic Regression has the best accuracy, precision & recall compared to others (Hindistan et al., 2019).

Chen (2021) studied Credit default risk prediction of lenders with resampling methods. The author analyzed the advantages of peer-to-peer lending to traditional banking due to the evolving technology in finance. He stated the challenges facing Peer-to-peer lending to be both parties not having adequate information about each other. The objective of this paper was to give a default risk prediction to forecast the probabilities of borrowers defaulting, in order to improve the company's decision-making. This paper uses SMOTE, NearMiss, manual 1:1 random selection, and XGBoost to predict if lenders can pay back on time. However, sampling methods such as LightGBM were used to predict credit default risk. The result shows that LightGBM with SMOTE performs best, using different classifiers to help researchers improve in default risk of prediction (Chen, 2021).

Shih et al. (2019) studied early warning systems in volume burst risk assessment of stock with big data platform. The author discussed the relationship between trading volume and stock returns. He further explained the factors affecting the trading price fluctuations in the stock market. The study's goal was to assess real-time stock trading volume according to the bursting complete index of trading volume and provide risk notifications using various trading volume levels utilizing real-time stream data processing architecture in the Big Data Spark framework. The author used different modules for data collection, Big data analysis, Decision module, process module, and message modules. The data source was gotten from the Yahoo stock market website. The datasets used were divided into training and testing data and online data validation. The top 50 stocks of market data were used and a total volume of 11,270,035 transactions was made. Predictors such as stock prices, daily opening, closing, trading, buying prices, and selling prices were used for prediction. The decision-making module was based on the rockburst risk rating. The result shows that investors used the high-risk stock market to predict the fluctuating risk of stock volume (Shih et al., 2019).

Ananthu et al. (2021) studied Credit Card Fraud Detection using Apache Spark Analysis, the rate of fraud has greatly increased due to the high dependence on internet technology. The financial sector has been greatly hit by these fraudsters, through the use of credit cards and internet banking. The objective of the author's study was to identify fraudulent transactions by analyzing the previous set of transaction records using big data analytics with machine algorithms for fast detection of large real-time data. The method used implemented an algorithm to help machine learning techniques. 100GB of datasets from various sources was gotten, which consist of 30 features. The yarn was used as the cluster manager. A supervised learning method was used to find the regression models. The result shows that apache spark is a better approach compared to Hadoop, Flink, and MapReduce. The author also compared different machine learning algorithms such as the decision tree classifier, logistic regression, and random forest (Ananthu et al., 2021).

Armel and Zaidouni (2019) studied Fraud Detection Using Apache Spark. Fraud is a continuous and ever-growing issue in the banking sector. fraud detection allows us to identify the fraudulent activities of cyber criminals in the banking sector. The objective of the study is to use the apache spark framework to estimate the performance of card fraud detection. (ARMEL et al., 2019), compared four different algorithms namely: simple anomaly detection algorithm, decision tree algorithm, Random Forest algorithm, and Naïve Bayes algorithm. A randomly generated dataset (Brownlee, (2020))was used for the simulation comprising of price and distance. The performance was analyzed based on the total running time and the accuracy of the above algorithms. The outcomes demonstrated that the simple anomaly detection algorithm produced the worst outcomes while the random forest method produced the greatest outcomes (Armel & Zaidouni, 2019).

Zhou et al. (2019) studied the Big data mining approach of PSO-based BP Neural network for financial risk management with IoT. Technology has recently been applied using IoT in the finance domain which generated data even in real-time. With The outcomes of the vastly expanding financial data from various sources, traditional models and neural networks might not be sufficient for credit risk assessment. A big data mining approach using PSO based Back-propagation neural network for financial risk management is proposed, for the construction of a large-scale nonlinear parallel optimization model by training, validating, and testing the dataset obtain. The evaluation was done of on-balance sheet and off-balance sheet items using Apache Spark and Hadoop HDFS. The outcomes of various experiment groups demonstrate that the suggested approach is highly accurate and capable of predicting financial risk as well as discriminating the default sample. Model testing and training take much less time to perform when using a huge data-parallel framework (Zhou et al., 2019).

Shivanna and Agrawal (2020) studied the prediction of defaulters using machine learning on Azure ML. They emphasized the need for banking systems to evolve into having their own credit risk assessment system using Basel II guidelines. The failure of the banking system to accurately predict loan defaulters has resulted in them acquiring huge amounts of losses. The objective of the research was to build different models on a dataset to determine which of the model would best predict loan defaulters. Datasets, comprising 25 attributes and 30,000 instances used on the machine learning tools such as Bayes Point Machine (BPM), Support Vector Machine (DSVM), Boosted Decision Tree (BDT), and Averaged Perceptron (AP) to predict loan defaulters. The results show that among the four methods used, the DSVM has the best prediction and was extended to the banking sector for implementation (Shivanna & Agrawal, 2020).

Berrada et al. (2022) analyzed the different applications of big data in the financial industry such as customized service, customer relationship management, segmentation, fraud detection, and credit risk assessment. The objective of the research was to use artificial intelligence for credit risk assessment. The author used different classification and prediction methods through data mining, supervised and unsupervised machine learning algorithms, and artificial neural networks to rework his data set to get an accurate prediction. He proposed that banks can become smarter, better, and quicker services and reduce loss due to credit defaulters. He used different machine learning algorithms such as support vector machine, decision tree, cat boost and logistic regression on the data. In conclusion, he noted that the classification method was best in predicting loan defaulters in other to get better accuracy and precision (Berrada et al., 2022).

Zhen and Wenjuan (2016) studied the commercial bank credit risk assessment method based on improved SVM. He proposed the use of a support vector machine ensemble for efficient classification performance using a fuzzy integral. The author built a model based on fuzzy integral Svm in other to train the data sets, bagging was done by using random samples on several examples. 16 indicators on 157 sample data were used to reflect the enterprise's profitability, efficiency, and debt-paying ability. He criticized domestic credit evaluation to have lesser data, statistical methods to have a bad effect, and neural networks as having difficulty getting better learning effects. The results showed that the Support vector machine as a classification method gives the best performance in predicting credit risk (Zhen & Wenjuan, 2016).

Li et al. (2018) studies the overdue prediction of bank loans based on LSTM-SVM. He noted that traditional bank loan risk prediction models such as KNN, Bayesian, and DNN are not effective in big data growth. He then proposed analyzing the dynamic behavior of the users using the LSTM algorithm to analyze borrowers' static data to solve problems with predictions. The author used the borrower's basic information to determine if the borrower was fraudulent. The author used the ability of LSTM-SVM to determine the fault tolerance and noise neuron as an advantage for his modeling. He further analyzed the algorithm model to train and test the data in other to determine the accuracy of each model. The result showed that the LSTM-SVM gave a better accuracy compared to the traditional methods (Li et al., 2018).

Misheva et al. (2018) studied network-based models to improve credit scoring accuracy using big data. He stated that the advancement in technology has significantly increased the risk faced in the financial sector and this has brought about different methods of approach to reduce the risk. He proposed a model for credit risk of peer-2-peer lending using correlation network-based models to improve the accuracy of the model and classify the risk type in p2p lending system. By using both supervised and unsupervised models to find the correlation network based on the solvency indicator. The dataset used consists of 727 European-based SMEs within a duration of 8years. Different machine learning techniques were performed using logistic regression with network parameters and also without network parameters and a CART method. The author concluded that loan default is due to the degree and closeness and centrality of predictors. The higher the degree, the higher the probability of default, which shows a positive correlation (Misheva et al., 2018).

Wu (2022) studied Real-time predictive analysis of loan risk with intelligent monitoring and machine learning technique. The author emphasized that Before granting a loan, financial institutions must evaluate and determine the potential default risk of a borrower. The risk and reward can be decreased and increased, respectively, by using mathematical models and algorithms to forecast the probability of loan default. This study employs the random forest method to create a loan default prediction model based on historical loan data from banks and other financial institutions. The random forest method was used to calculate the AUC which gave a value of 0.86, better than other methods such as decision tree and logistic regression with 0.80 each respectively. The experimental results show that this method surpasses other classification algorithms in terms of accuracy and recall rate, including decision trees and logistic regression. The random forest machine learning technique is also used to establish the dataset's feature rankings (Wu, 2022).

2.2 Study 2 – Email Classification of Text data using Machine Learning and Natural Language Processing Technique

Nandhini and KS (2020) analyzed the effectiveness of various machine learning algorithms for email spam detection. During the investigation, it was observed that a significant volume of unsolicited emails posed potential threats to users and their organizations' security. The primary focus of the study was to leverage established algorithms to build a machine learning model capable of classifying emails as either spam or ham. The dataset utilized for this research was sourced from the Spam Base dataset available in the UCI Machine Learning repository. Several machine learning methods for data classification were employed, including Logistic Regression, Decision Trees, Naive Bayes, KNN, and SVM. To assess the research findings comprehensively, performance metrics such as Classification Accuracy, Confusion Matrix, Precision, Recall, and F1 Score were utilized. Among the five machine learning models investigated, the Random Tree model exhibited the highest performance index when applied to the UCI Machine Learning dataset. The results underscored the Random Tree model's exceptional performance, achieving accuracy, precision, recall, and F1 Score scores of 99.9%, 99.9%, and 99%, respectively. (Nandhini & KS, 2020)

Kumar and Sonowal (2020) studied machine learning methods for email spam detection. He clarified that emails are deemed spam when they are delivered to a large number of recipients in an unsolicited manner or with an advertising message attached, wasting space, time, and transmission speed. He advised using machine learning to detect spam as an efficient method. In his method, the author first preprocessed the data by eliminating any missing values from the data set before utilizing text analysis to evaluate email messages. Using several machine learning traditional classifiers, ensemble learning techniques, boosting and adaboost classifiers, data transformation, data reduction by deleting stop words, tokenizing each word, and producing a bag of words for feature selection are all steps in the process. Naive Bayes, Support Vector Machine, Decision Tree, K-Nearest Neighbor, Random Forest, Adaboost Classifier, and Bagging Classifier are some of the machine learning methods employed by the researcher.

The algorithm with the highest accuracy, 98%, according to the results, is the Multinomial Naive Bayes classifier Kumar and Sonowal (2020).

Junnarkar et al. (2021) employs machine learning and natural language processing to classify spam emails. He noted that the amount of spam communications has rapidly expanded with the rate of information transmission via email. In order to lower the rate of unsolicited bulk emails or spam, the research underlined the need to develop a thorough system for spam classification based on semantic text classification utilizing NLP. As part of the innovation of the work, the researcher also categorizes the URL present utilizing a three-step filtering and analysis procedure. The dataset was first preprocessed using various text classification approaches, and once it had been trained using machine learning algorithms, URL filtering was carried out to safeguard users from emails containing harmful URLs. The method was applied to two distinct datasets, namely the Enron spam dataset and spam.csv from Kaggle. Utilizing feature engineering, text was preprocessed using naive bayes, KNN, decision trees, random forests, and support vector machines, among five other machine learning algorithms. The most effective machine learning algorithm, according to the results, was support vector machines, which had performance indices of 91% accuracy, 94% precision, 93% recall, and 91% F1 score. (Junnarkar et al., 2021)

Olatunji (2017) examined the significance of spam detection as a first stage in the email filtering process in the wake of an Increased influx of unsolicited messages, which now make up the bulk of inbox messages. His work focuses on the application of support vector machines and extreme learning machine models for email spam identification. He found that there has been a lot of study comparing ELM with SVM for classification and regression difficulties. The author focused on the usage of SVM and ELM for email spam detection and the requirement for an efficient detection method to identify and separate unsolicited mail. His study's findings demonstrated that the SVM provided a better performance index than the ELM, with an accuracy score of 94% compared to 93% for the latter. The ELM did well in terms of speed, with a training time of 0.94 seconds, while the SVM fared better in terms of accuracy, despite the fact that both models produced good results. He also contrasted the speed of the operation of the two models. (Olatunji, 2017)

Feng et al. (2016) explored support vector machine-based naive bayes algorithm for spam filtering. In his paper, he highlighted the use of naive bayes classifiers for spam emails and pointed out their shortcomings, such as having strong independence assumptions across features. A naive bayes-SVM-NB filtering system was presented in the study as a spam detection method. A hyperplane was used to divide the training sample into two sections after the classification and training were first carried out using naive bayes classifiers. The training set was reduced and the independence of the samples in each category was improved by using the trimming approach offered by SVM to remove poor examples from the training space. When compared to a pure SVM-based solution, the results indicated that SVM-NB delivers improved precision and recall rates for detecting spam (Feng et al., 2016).

Verma et al. (2020) characterized phishing as a form of network theft in which attackers produce phony web pages or websites that seem quite legitimate with the intention of tricking unsuspecting users. He continued by saying that emails are becoming a very popular medium where attackers transmit nefarious links or pop-up ads that recipients unintentionally open, leaving them entirely exposed. The goal of the study is to compare the various accuracy rates and provide a full description of how phishing is classified using machine learning and natural language processing techniques. For better results, various data preprocessing techniques were used, including the removal of stop words, punctuation, tokenization, and stemming. The results demonstrated the various machine learning algorithms used for classification, including K Nearest Neighbors with accuracy of 94.75, Decision Tree with accuracy of 97.55, Random Forest with accuracy of 98.42, Logistic Regression Classifier with accuracy of 98.56, SGD with accuracy of 98.34, Nave Bayes classifier with accuracy of 98.70, and SVM Linear classifier with accuracy of 98.77. The SVM linear is recommended as the best ML algorithm for the used dataset because it had the best accuracy (Verma et al., 2020).

Fette et al. (2007) used machine learning to study phishing email detection. They stressed that emails with links to websites that gather information are the most prevalent way for phishing attacks to start. The report noted that because hackers are becoming more skilled every day, it is crucial to recognize phishing assaults. The study's goal was to use machine learning to identify email phishing assaults. 860 phishing emails and 6950 non-phishing emails made up the dataset. Over 96% of phishing emails were successfully recognized by the model, with only 0.1% misclassified. The technique employed was a machine learning strategy known as PILFER, including decision trees, random forests, SVM rule-based strategies, and Bayesian as the classifier. The difference was not statistically significant, according to the researcher's analysis of total accuracy. The result shows that the PILFER approach gave an overall accuracy of 99.5% (Fette et al., 2007).

Bahgat et al. (2018) noted that monitoring and categorizing a large number of emails is a significant difficulty since there has been a constant increase in email users, which has led to unsolicited emails. They devised an effective email classification strategy based on semantic methodologies. They established a productive email filtering system based on semantic techniques. The approach taken in his research uses the wordnet ontology together with several semantic-based approaches and similarity measurements to reduce the complexity of textual properties over time and space. Various feature selection methods, including PCA and correlation feature selection, were employed. To test the accuracy, the researchers employed a variety of machine learning techniques, including logistic regression, naive bayes, support vector machines, random forests, and radial basis function networks. They also demonstrated that, when compared to other machine learning techniques, logistic regression provided the highest accuracy. He showed that, with the addition of feature selection techniques, the outcome provided an average of 90% accuracy with shorter execution times (Bahgat et al., 2018).

Toolan and Carthy (2010) described methods for phishing and spam feature selection. Unsolicited Bulk Email (UBE), as the researcher called spam emails, continuously evolves and gets past some junk mail filters, which is why his study is important. The researcher looked into the usefulness of 40 traits by calculating information gained over spam, phishing, and ham corpora. The results with the highest information gain created the best classifier after each dataset's evaluation was tested using its information gain. With the help of the C5.0 decision tree learning method, each dataset attribute was assessed. The findings demonstrated that the classifier developed using the best features outperformed those created using the best IG values, median IG values, and lastly the worst IG values (Toolan & Carthy, 2010).

Magdy et al. (2022) described a deep learning-based spam and phishing email filtering method. He stated that spam emails pose a serious threat to the internet, consume a lot of server storage space and cause delays in surfing unwanted bulk emails (UBE), which ultimately costs people and businesses money. A deep learning model that performed better than comparable experiments was presented in the research report. The research's classifier supported three different classes: phishing, spam, and ham. A predictive ANN model with two hidden layers and a tolerable training time was introduced. The outcome compared the accuracy of deep learning ANN to that of machine learning random forest and SVM. When compared to conventional machine learning models, the deep learning model's accuracy was 99.9% (Magdy et al., 2022).

Zhang and Yuan (2012) employed a neural network to identify phishing in emails. In light of the increasingly complex attacks, they emphasized the significance of efficient email filters for phishing emails. The goal of the research was to identify phishing emails using multilayer feedforward neural networks. Large real-world samples of 4202 spam emails and 4560 phishing emails make up the dataset used. The technique adopted included feature selection, phishing dataset processing, neural network system implementation, and performance comparisons between NN and machine learning techniques. The outcome compares the performance of the neural network with five machine learning models for the dataset in use. The researcher claimed that while the decision tree algorithm had the highest accuracy (96%), neural networks had the highest recall (95%+ precision). NNs are therefore more effective at spotting phishing emails, as stated in the study (Zhang & Yuan, 2012).

Mujtaba et al. (2017) utilized machine learning to study email classification for forensic analysis, focused on the impact of an enormous increase in email data and the difficulties in email management. He emphasized the need for email detection and classification based on content and other data elements. The strategy suggested using multiple labels to classify emails in order to arrange them and aid in the forensic analysis of large amounts of email data. With the use of a unique blend of TF-IDF properties, the emails were divided into four classes. Five distinct machine learning algorithms—logistic regression, naive Bayes, stochastic gradient descent, random forest, and support vector machine—were used to assess the effectiveness of the technique.   Comparing several machine learning methods used for classification, the logistic regression method produced the highest accuracy. With an accuracy of 91.9%, the outcome demonstrates that logistic regression outperformed alternative techniques (Mujtaba et al., 2017).

Bagui et al. (2021) examined the effectiveness of deep learning and machine learning for classifying phishing emails using one-hot encoding. He stressed the significance of creating an anti-phishing technology to aid people and companies in avoiding losing enormous sums of money to criminals. To determine the characteristics of the text body, the researchers used a semantic analysis. Phishing emails made up 3,416 of the 18,366 tagged records in the sample, while regular emails made up 14,950. Data cleansing, lemmatization, and vectorization utilizing one-hot encoding were some data pretreatment techniques employed to get the dataset ready for machine learning and deep learning model. 30% of the dataset was utilized to validate and test the model's performance, while 70% was used for training. The Convolutional neural networks (CNN) and long memory DL techniques were applied. (LSTM) (Bagui et al., 2021).

Chakravarty and Manikandan (2022) proposed clever techniques for identifying email spam using machine learning techniques. He underlined the importance of an efficient anti-spam filtering system and described spam as one of the greatest threats to contemporary internet usage. The authors stated that one of the largest issues in a supervised learning system is anticipating the class labels in a personalized mailbox; as a result, a reliable detection system is required. The research methodology included an NLP component that can separate spam from non-spam emails and further categorize them. On the SpamAssassin spam corpus dataset, various well-known techniques, including Bayesian classification, KNN, ANN, SVM, counterfeit safe framework, and unpleasant sets, were applied. The naive bayes gave and the unpleasant sets performed adequately well according to the results of the six machine learning algorithms utilized, but the NB gave the best performance index when compared to the other methods. Naive Bayes provided accuracy, precision, and recall values of 99.48%, 99.68%, and 99.48% respectively. The algorithm's output was then put into practice, and the potency of the categorization method was established. An overall dataset of 6000 spam and non-spam emails is used by the author (Chakravarty & Manikandan, 2022).

2.3 Study 3 – Loan Prediction using Azure Machine Learning

Alshouiliy et al. (2020) stressed the importance of using contemporary technology, such as machine learning algorithms, by lenders to examine and anticipate loan borrowers' creditworthiness. His choice of using Azure platform was because azure provided good features due to its easy to create, deploy, manage platforms. The research focuses on predicting the possibility of customers' loan repayment using Azure machine learning techniques. The lending club dataset was used to model the business challenges which covers from years 2017 to 2018. The researcher compared the performance of two Azure machine learning algorithms namely Two jungle algorithms and two decision tree techniques the algorithm was used to train the dataset, and finally model evaluation was done using different performance indexes such as accuracy, precision, recall, F1, and AUC to get the result. The results show that the jungle forest performed better with a higher performance index than the decision forest because it allowed tree branches to merge. (Alshouiliy et al., 2020)

Shivanna and Agrawal (2020) stated the importance of Azure ml in predicting defaulters in the financial sector, they noted that businesses and financial organizations lose large amounts of money due to their inability to accurately predict loan defaults in a timely manner hence leading to bad debt. The researcher's goal was to evaluate how well different ML classification algorithms predicted credit card defaulters. 25 attributes and 30,000 attributes from the machine learning repository make up the dataset used in this study. To create various models for better prediction, this research used a variety of machine learning methods, including the Deep Support Vector Machine (DSVM), Boosted Decision Tree (BDT), Averaged Perceptron (AP), and Bayes Point Machine (BPM). The machine learning model was trained and evaluated, and its performance was assessed using a variety of performance indices. For model evaluation, metrics like accuracy, precision, recall, F1score, and AUC were utilized. With an accuracy of 82%, precision of 69%, F1-score of 47%,, AUC of 74% and recall of 36%, the Deep Support Vector Machine outperformed the other four machine learning algorithms utilized. (Shivanna & Agrawal, 2020)

Motwani et al. (2018) suggested predicting the worthiness of bank customers using machine learning over the cloud. He defined creditworthiness as the “probability of default” on loans by financial institutions. He noted that financial institutions generate a vast amount of data in volumes, velocity, and variety hence the need to harness it, he stated the effectiveness of cloud computing in addressing financial data. The objective of the research was to build and access the performance of 3 machine learning algorithms for predicting defaulters using Azure machine learning platforms and propose the predictive analysis framework for classifying and predicting payment defaulters. The dataset used was obtained from the UCI repository in order to detect creditworthiness, on the large dataset different machine learning algorithms were used. They developed a neural network-based model on Machine Learning algorithms. The model built above was tested over the "Microsoft Azure Cloud" platform. To choose the appropriate variable to utilize, several data preprocessing and feature selection were applied to the dataset. The dataset was then divided into the training and validation groups. The researcher's suggested approach was put to the test by the author, who contrasted it with well-known machine learning techniques including Bayes point, logistic regression, and decision tree. The outcome demonstrates that the suggested neural network-based model beat the three widely used machine learning methods tested by the author, with accuracy and recall of 82% and 41%, respectively..(Motwani et al., 2018)

Arun et al. (2016) studied a machine learning approach to loan prediction in the financial sector. The researcher described how droughting and risky loan prediction could be to financial institutions and decision makers and further suggested an approach to effectively predict loan defaults. The primary objective of this research is to predict how safe it is for banks to issue out loans using certain conditions such as data mining the big data of historical data of loan defaulters. The dataset used consisted of thirteen attributes with 981 records with both numerical and categorical data. The researchers explained the data collection process, comparing different machine learning models, training of the models and finally testing the model performance. The data was further trained using Various machine learning approaches such as logistic regression and random forest to determine the performance of the model on the dataset. The result was experimented using HTML, CSS, and Django to show the loan prediction system using different attribute. (Arun et al., 2016)

Shoumo et al. (2019) researched the application of machine learning in evaluating credit risk in financial institutions. He found the difficulties in financial institutions effectively forecasting loan defaulters, which results in substantial loss. In the article, it was suggested that credit lending organizations use machine learning models to accurately analyze and forecast loan defaulters. Machine learning techniques like SVM, Random Forest, Extreme Gradient, and Logistic Regression were used to conduct various comparison analyses. The dataset used contains information about individuals who have applied for loans in the past. Dimensionality reduction was carried out using Recursive Feature

Elimination and Principal Component Analysis. AUC score, accuracy, recall, and other performance indices were used to analyze the model's performance. The support vector machine and recursive feature removal are two of the various machine learning models taken into consideration. (Shoumo et al., 2019).

Tumuluru et al. (2022) conducted some comparative analysis of customers loan approval prediction using machine learning and noted that financial institutions could use machine learning methodologies rather than credit scoring and risk assessment procedures for loan projections. The researcher wanted to utilize machine learning to estimate future loan defaulters by extracting patterns from a dataset of loans that had been approved. The training set contained 70% of the dataset, and the testing set contained 30%. Four machine learning algorithms were used to train the dataset: random forest, support vector machine, K-nearest neighbor, and logistic regression. The evaluation of the results revealed that random forest outperformed the other algorithms with an accuracy of 81%. (Tumuluru et al., 2022)

Kachhwaha and Shrivastava (2020) noted that financial companies face a hurdle when determining threat elements to take into account when granting loans or credit to customers. He went on to list the disadvantages of machine learning prediction, including data privacy and data inadequacy. By contrasting various machine learning techniques, the research's goal is to develop a predictive model for credit threat identification based on an ensemble machine learning methodology. Microsoft Azure ML studio was utilized to carry out the predictive classification technique. It used two class bayes, two class logistic regression, and a proposed model in addition to the usage of features selection utilizing the cloud-based dataset. The result of the azure machine learning studio gave the proposed work based on deep support vector machine as the model with the highest accuracy, hence was the preferred model for the dataset used. (Kachhwaha & Shrivastava, 2020)

Addo et al. (2018) examined utilizing deep learning and machine learning algorithms for credit risk analysis. They emphasized the significance of credit risk as being essential to transparency and decision-making. The study's goal was to create a binary classifier for estimating the likelihood of loan default based on machine learning and deep learning models. 10 important feature selection was conducted on the dataset before it was used in the modeling process. The dataset consisted of 117,019 lines representing the probability of default from customer’s past record. A SMOTE technique was used to obtain a balanced dataset with a ratio of 46% for non-default and 54% for default. Different machine and deep learning algorithms were used such as Random Forest, logistics regression, Gradient boosting machine, and neural network. Performance index for company’s credit worthiness such as ROC, AUC, RMSE was used. The result shows that the tree-based model performs better than the multilayer artificial neural network. (Addo et al., 2018)

Coşer et al. (2019) outlined predictive algorithms for calculating the likelihood of loan default. According to the researcher, determining a borrower's creditworthiness will always be necessary given the expanding number of loan requests. He went on to explain how data scientists' job is to find the insights that explain customer behavior and profile. He offered an approach for predicting loan default that makes use of data mining and machine learning algorithms. The methods discussed in this study evaluate the likelihood of default using classifiers like lightGBM, XGBoost, Logistic Regression, and Random Forest. He contrasted how datasets that were imbalanced and those that were balanced were classified, and he offered a model based on how well they performed. Performance indexes such as AUC score, Precision, Recall and Accuracy was used to evaluate the model. The result shows that the Random Forest classifier gave the best outcome with an AUC of 89%. (Coşer et al., 2019)

Padimi et al. (2022) suggested applying machine learning technique to maximize the performance of loan default prediction. His research was focused on peer-to-peer lending which involves borrowing from P2P platforms where borrowers can access loans at much lower interest rates than the traditional lenders. He further stated the risk inherent in such ventures where borrower may default on loans. This paper addressed the issue of loan default in peer-to-peer lending platforms thereby encouraging lenders to continue providing loans. The dataset considered had 112 attributes including missing values, different data preprocessing techniques where performed such as data cleaning, feature engineering, dimensional reduction, data scaling and transformation before training and testing the model. To categorize loan data and estimate the chance of default, the researcher took into account 5 machine learning algorithms: Logistic Regression, Naive Bayes, Random Forest, K-Nearest Neighbor, and Decision Tree. Different cross-validation results and performance indices are shown as a result. The Random Forest ML method had the highest accuracy, at 94.54%.(Padimi et al., 2022)

Koç and Sevgili (2020) employed a predictive algorithm to find the first consumer loan payment with a focus on finding clients early in a credit facility. The goal of this study is to use predictive analysis to examine consumer behavior in relation to loans when a customer's first payment is late. 598,669 rows and 45 columns in almost 600,000 records from a Turkish bank database system. The dataset was a significantly unbalanced two-class dataset with 99.5% FPD and 0.5% non-FPD. In order to address the imbalance dataset, the study evaluated under sampling and oversampling techniques. On the oversampled and under sampled dataset, four distinct machine learning techniques—Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest were applied. The result of the two-class random forest using under sampling yielded 86% compared with other ML algorithms used to measure its performance index. The performance metrics used in the research are accuracy, precision, recall, f1score. (Koç & Sevgili, 2020)

Zhu et al. (2023) presented a machine-learning-based prognosis of loan default that was comprehensible. They described how a machine learning prediction model was applied to online loan platforms. In order to increase customer confidence in ML technology, their research examined the need to forecast how machine learning principles and models can be further communicated. They proposed a loan default prediction model based on 4 distinct ML concepts, decreased the number of feature dimensions using approaches including deletion, PCA, and feature interaction, and suggested the best prediction based on its performance index. Different machine learning models' performance, including that of LightGBM, XGBoost, decision trees, and logistic regression, was examined. The prediction outcome demonstrates that the LightGBM and XGBoost models outperform logistic regression and decision tree models in terms of performance. Accuracy, Area Under Curve, and precision performance indices all had values of 72%, 80%, and 55%, respectively. The prediction results were subjected to explainable research utilizing the local interpretable model-neutral explanations technique. The results show how factors like loan length, loan grade, credit rating, and loan amount affect the expected outcomes.. (Zhu et al., 2023)

CHAPTER 3

3 STUDY 1- EFFECTIVE CREDIT RISK MANAGEMENT IN FINANCE USING BIG DATA TECHNOLOGY.

3.1 Introduction

Financial institutions are faced daily with high rates of non-performing loans because of the inability to accurately predict defaulters before the disbursement of loans, this has led to many failures and the eventual collapse of some financial industries particularly the banking industries. The bank's profitability depends on how well it's able to minimize nonperforming loans called bad debt. Through effective credit risk management, banks can predict, analyze, and mitigate the possibility of non-performing loans, which would greatly improve their profitability. Predicting loan repayment in the past has been a very exhausting task which has been greatly done through traditional methods such as calculated risk analysis through credit scoring and risk profiling methods, this has not effectively prevented and predicted risk because of the dynamic nature of defaulters.

With the large volumes of data generated by the financial sector, there has been a need to effectively give meaningful insight to those datasets for business development and continuity. Big data analysis can be used in the financial sector to analyze customers' behavior, and credibility by effectively predicting the tendency of customers to default using their previous financial records. A dataset can be considered big data if it has the following characteristics such as Volume, value, velocity, variety and velocity. Big data can be generated in a structured, unstructured, and complex format from multiple sources. To effectively manage risk different machine learning algorithms can be used to find the relationship and categorize different datasets to give a meaningful insight and guide the decision-making of an organization.

The objective of this research is to reduce the level of non-performing loans in the banking industries, by effectively predicting the probability of default using past customer records such as personal income, loan amount, interest rate, default rate, loan purpose, loan intent, loan status etc, to give a meaningful insight using different machine learning algorithm for proper decision making. This model would serve as a guide to determine customers' behavior toward loan repayment and further categorizes customers into defaulter and non-defaulter by finding the relationship between each independent variable. For business continuity, a proper understanding of business models through data analysis is imperative for business sustainability and competitive advantage. Financial institutions can significantly maximize profit and establish a good borrower to lenders relationship by mitigating credit risk using big data. The traditional method of taking the calculated risk from customer profiling can be greatly eradicated with the emergence of big data technology using machine learning algorithms. The term "machine learning techniques" refers to a variety of software tools that may turn data sets into "models," which are able to represent the data set and generalize it in order to make predictions about new data. Machine learning models are generally classified into dependent variables and independent variables. In general, the independent variables are the many elements that influence the outcome of the prediction and are used as the model's input. The dependent variable refers to the target of the prediction and is used as the output of the model.

The prediction of loan defaulters can be categorized as a classification problem, the specific task called the class label was the loan status, and the machine learning algorithm predicts from the given dataset and divides it into a training dataset, which would be used to train itself. The training data comprises all different scenarios of the problem to enable proper training. Machine learning uses artificial intelligence to learn from a given set of data. ML is classified into supervised, semi-supervised, and unsupervised learning. In this paper, Apache spark was used as a processing engine and HDFS was used to store the dataset, the dataset was further classified into test_train_split models using machine learning algorithms on a supervised learning technique. The dataset used constitutes both numerical and categorical data using different machine learning algorithms such as Decision Tree, Random Forest, and Naïve Bayes to help make predictions by analyzing the inputted dataset, learning from it, and using the result to improve better decision-making. The binary classification techniques gave two classes of labels: the output normal state was assigned a value of 0, and the abnormal state was assigned a value of 1.

The results presented different machine learning outputs from Decision Tree, Random Forest, and Naïve Bayes. The study attempted to identify the most suitable algorithm by considering accuracy, recall, and precision. However, it's important to note that there isn't a universally superior model, as performance varies depending on the input dataset's characteristics.

3.2 Methods

A data pipeline is used to explain the methods, which shows the sequence of steps in arriving at a successful prediction result. Credit risk management is a classification problem where the financial institution is determining whether customers would default or not in the payment of a loan. It’s either the customer defaults called 1 or his non-defaulter called 0. Big data can be used to predict a long range of problems such as cancer detection, virus attack detection, and financial prediction. The dataset is first stored in the HDFS to enable the machine learning process, then a comma-separated values file (CSV) is imported on the spark environment using jupyter notebook, where further processing is done. A series of data preprocessing was done using Apache spark as the processing engine. The modeling was done using machine learning algorithms such as Decision Tree, Random Forest, and Naïve Bayes on a spark library such as Pandas for cleaning the data, NumPy used for mathematical calculations, and seaborn used for visualization. The proposed method would be used for early prediction of bank customers defaulting in repayment of loans based on machine learning algorithms and ensemble learning techniques. The model is composed of the following stages, Data Understanding, Data preparation, Data loading, Machine learning modeling, and evaluation. A Data pipeline flowchart is used to explain the different processes and stages of processing the given data as illustrated in fig 1.0 below.

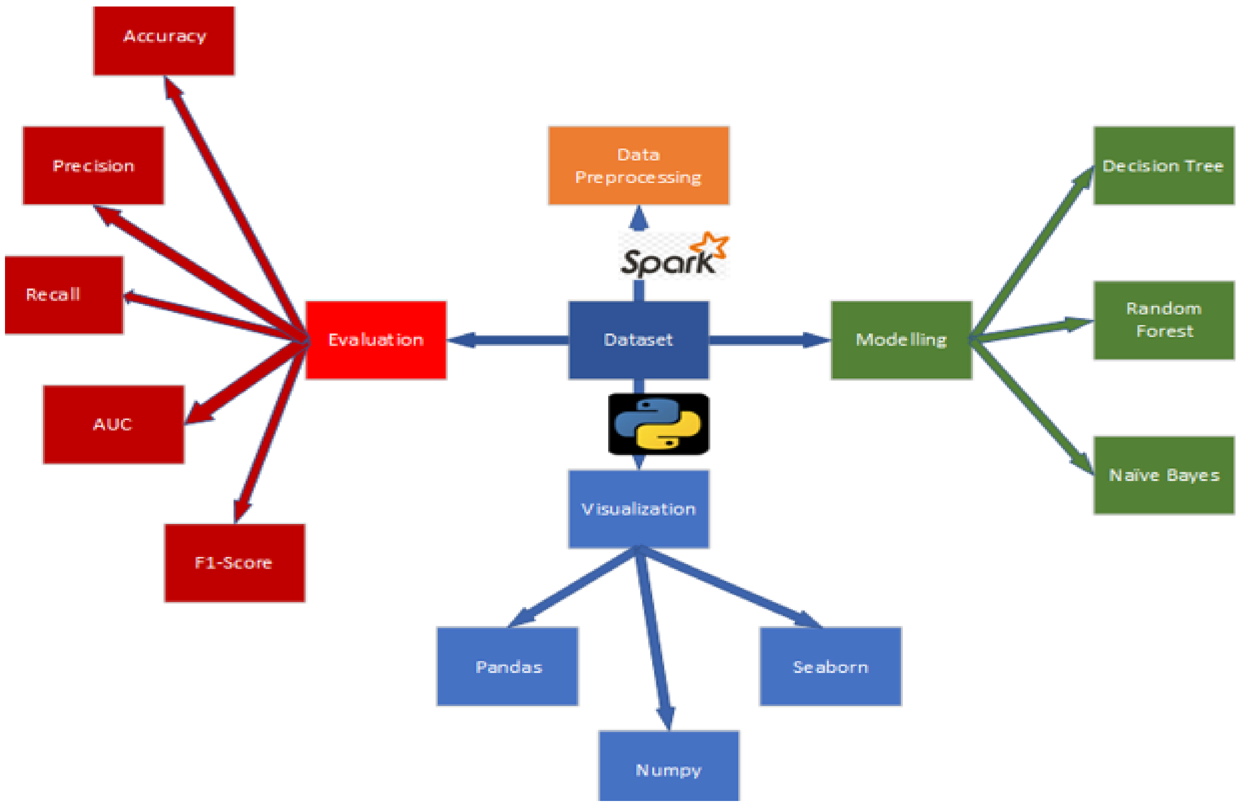


Figure 1 Proposed Methodology *(Shivanna & Agrawal, 2020)*

3.2.1 Data set

The data set shows a collection of related sets of information for analyzing credit risk in the financial sector, it is composed of separate elements from the stimulation of credit bureau data which would be used to predict information of a real-life scenario.

3.2.2 Data Understanding

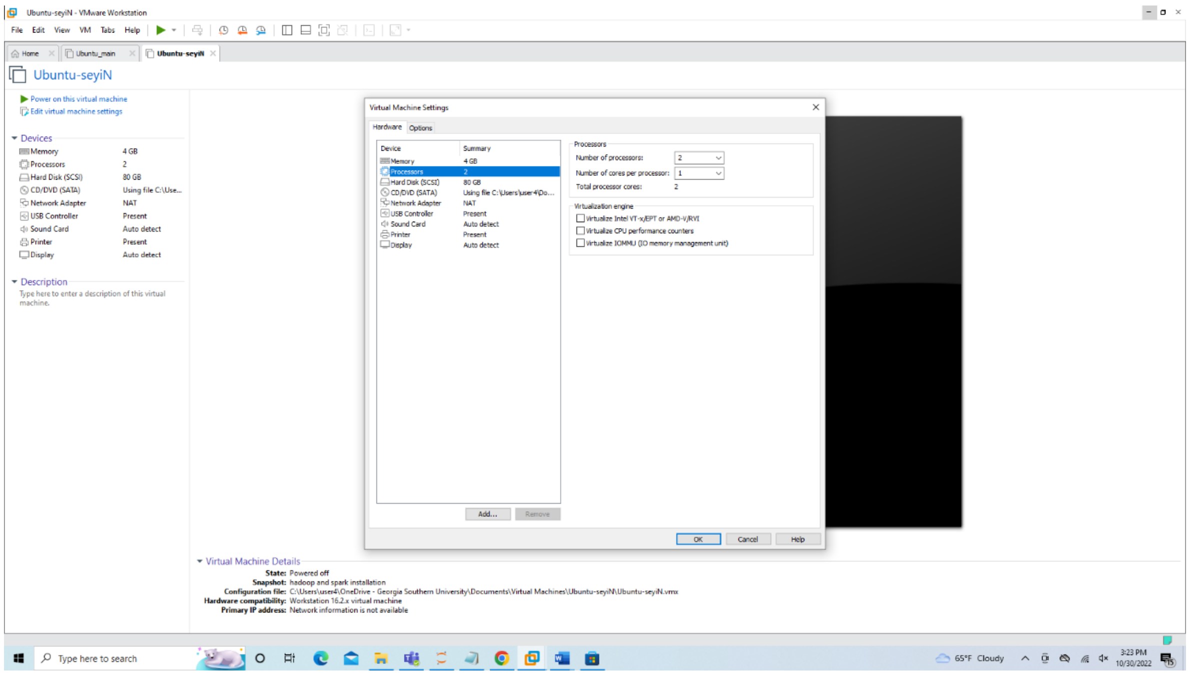
The dataset consists of 390,984 observations, with 12 columns and 32,582 rows. The data set has been concatenated into one big sheet of 1.763MB in a CSV format the dataset was gotten from www.kaggle.com. This different feature of the data shows the borrower's profile gathered in the year 2020. The research would be predicting the loan status, based on supervised learning.

3.2.3 Data Preparation

Some preprocessing steps were used to remove attributes of extreme values then the dataset was rescaled to be transformed into a single form that the machine learning model can understand. The data preparation and data cleansing were done to systematically remove void space in the dataset to avoid errors. The following preprocessing was done to the data sets, data cleansing & reduction, feature standardization, and normalization of the data set (Shaheen & ElFakharany, 2018).

3.2.4 Data Loading

The virtual machine used was VMWARE Pro 16, on an Ubuntu 22.04 LTS operating system. The virtual machine had a Master Node of 2 VCPUs on an 80GB hard disk memory. Then Hadoop-3.2.4 was installed for storing the large dataset, the file in HDFS was divided into many blocks, which were then stored in a collection of Data Nodes. The Name Node executes the file system namespace actions, such as opening, shutting, and renaming files and directories, among other specific tasks. The file is initially distributed into HDFS with a replication factor of 1 and a block size of 128MB. Figure 2.0 shows the virtual machine features used for allocating memory and disk type. Figure 3.0 also shows the different nodes in HDFS used for processing and storage.



Devices

Memory	4 GB
Processors	2
Hard Disk (SCSI)	80 GB
CD/DVD (SATA)	Using file C:\Use...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Description
Type here to enter a description of this virtual machine.

Virtual Machine Settings

Device	Summary
Memory	4 GB
Processors	2
Hard Disk (SCSI)	80 GB
CD/DVD (SATA)	Using file C:\Users\user4\Do...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

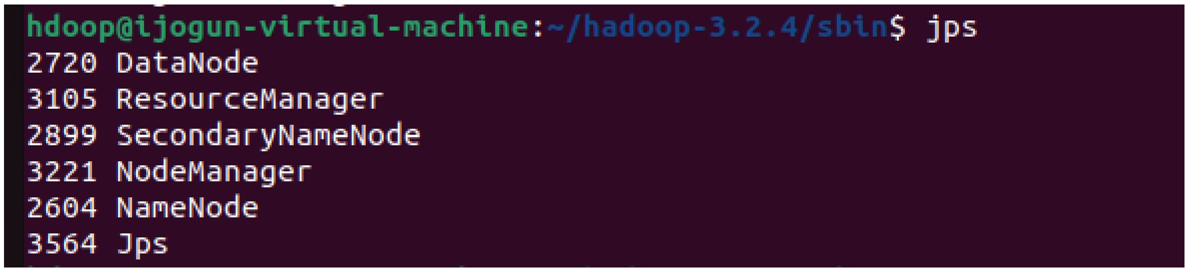
Processors
Number of processors: 2
Number of cores per processor: 1
Total processor cores: 2

Virtualization engine
☐ Virtualize Intel VT-x/EPT or AMD-V/RVI
☐ Virtualize CPU performance counters
☐ Virtualize IOMMU (IO memory management unit)

Virtual Machine Details
State: Powered off
Snapshot: hadoop and spark installation
Configuration file: C:\Users\user4\OneDrive - Georgia Southern University\Documents\Virtual Machines\Ubuntu-seyiN\Ubuntu-seyiN.vmx
Hardware compatibility: Workstation 16.2.x virtual machine
Primary IP address: Network information is not available

65°F Cloudy 3:23 PM 10/30/2022

Figure 2 Virtual Machine Setup



```
hdoop@ijogun-virtual-machine:~/hadoop-3.2.4/sbin$ jps
2720 DataNode
3105 ResourceManager
2899 SecondaryNameNode
3221 NodeManager
2604 NameNode
3564 Jps
```

Figure 3 HDFS Nodes in the Virtual Machine

Apache spark, an open-source framework used for big data analysis, has many advantages over Hadoop such as supporting in-memory computations of RDDs. Due to its fast-processing capacity, it can be used for real-time stream processing. It provided a high processing speed which is 100x faster in memory and 10x faster on the disk. Spark-3.3.1 was installed for processing the data set, and spark DAG which is the execution engine was used to facilitate the in-memory computation and acyclic data flow giving it high speed. Multiple languages such as java, R, Scala and Python are supported with spark. However, Spark using python an open-source spark library was used for processing the dataset in this project. PySpark was used to run python applications by relying on Apache spark capacity in the spark library. Figure 4.0 below shows the component of data pipeline.

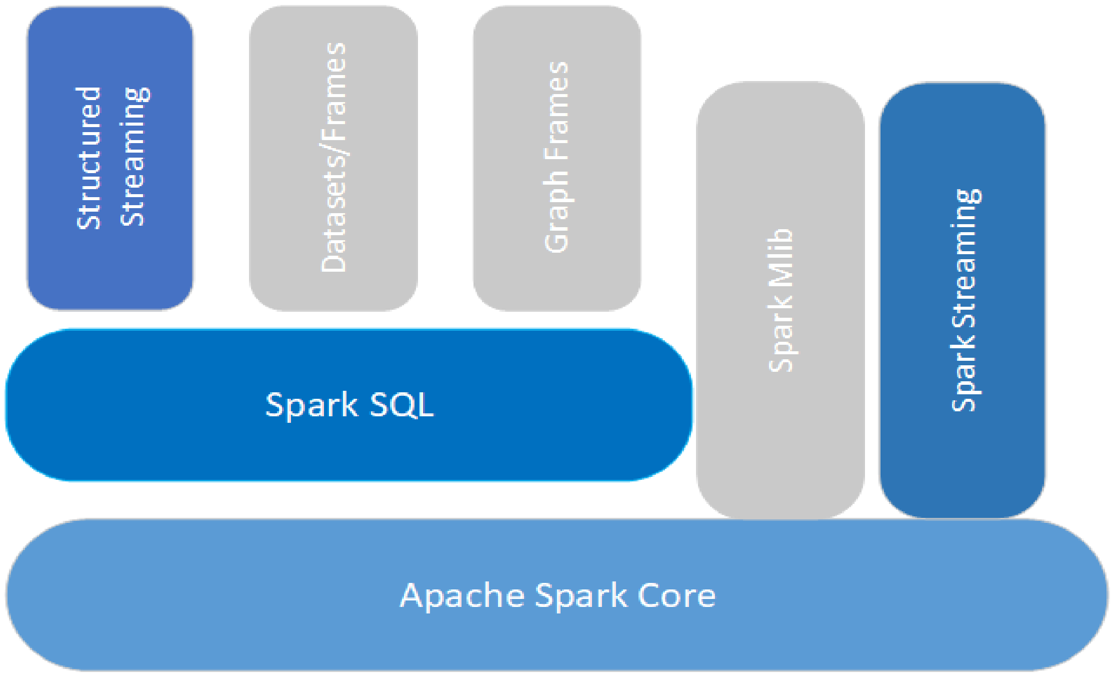


Figure 4 Spark Components

The data set was processed using a Jupytar notebook. The data set named credit_risk_dataset.csv was uploaded to the Master Node and saved into the HDFS. Then on Jupytar notebook, Python was selected to run the program to enable the dataset to be used for machine learning techniques. The goal is to apply Random Forest, decision tree, and naive Bayes machine learning classification on HDFS and use Spark for processing.

3.2.5 Machine Learning Classification Technique

The classification technique would involve using machine learning algorithms that would learn how to assign classes from a problem domain into a class model. The model used for my classification technique is multi-class classification which involves algorithms such as decision trees, Naïve Bayes, and Random Forest. (Brownlee, 2021). A supervised learning model was used to see how well the model would perform. A model validation process was used called the split-train and test model were used to train the dataset (BUENO, 2022). The first step in the machine learning process is to arrange the data into an acceptable split train model. In pyspark, the command to import the machine learning language is called the Scikit-learn where the data is separated into the “features” and “Targets”. Decision tree, Random Forest and Naïve Bayes algorithm are deployed from python’s scikit-learn library using pyspark.

The algorithm has been able to run on HDFS successfully through its remote access using the internal IP of the Master node. Below is the formula for regression, a machine-learning model used for prediction.

The linear regression model used for modeling the scalar response between one or more variables is usually a dependent and more independent variable.

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

Equation 1 linear regression
where y is the dependent variable; X is the independent variable and e is the error.

logistic regression model: this shows a relationship between a predictor variable and a categorical response variable.

$$p(x) = \frac{1}{1 + e^{-(x-u)/s}}$$ where u is the location parameter and s is the scale parameter

Equation 2 Logistic Regression Equation

3.2.5.1 Split data

The data set would be split into two pieces namely, the Training set and the testing set, by randomly selecting the samples of data without replacement. A ratio of 70% (0.7) of the rows would be put into my training set, and the remaining 30% (0.3) would be put into my test set (“X-train,” “X-test”, “Y-train”, “y-test”).

3.2.5.2 Train model

This involves building and training the model on the predictive model. It is denoted with “X_test” and “Y_test”). Then the model would be tested with (“X_test” and “Y_test”) and the performance is evaluated.

3.2.5.3 Evaluate Error

The predictive models were validated using the validation data set, this was used to fine-tune and check for overfitting of models. The performance of each model was evaluated using the confusion matrix. The confusion matrix is given the abbreviation True- Positive (TP), True Negative (TN), False positive (FP), and false Negative (FN).

3.2.5.4 Filter Values

The machine learning algorithm chooses a smaller part of the data set and uses it for analysis. In the research, a total of 8 attributes were used for the modeling.

3.2.5.5 Decision Tree

No algorithm can learn without any previous experience. A decision Tree as the name implies is like a flowchart in which the internal node represents a test, and the branches represent the results. They are the binary tree that recursively splits datasets, beyond leave nodes to only one set of numbers. It classifies the output in the form of a tree-like structure. It is used to process complex nonlinear patterns and predictions, it is a powerful and non-probabilistic technique that works well on many tasks and can handle numerical and categorical features, without scaling input data. The first node in the tree is called the root node, and an unknown class is labeled tuple T. decision nodes are known as the condition, and leave nodes is called the results. The decision continues until no further gain can be made.

The decision tree can be used to predict the class of a target variable. Different parameters in the machine library can be used to define decision tree such as maxDepth, maxbins, numClasses from the Scikit-learn library used by spark. The training complexity is defined by 0(nlogn)p) while prediction is o(p).

Entropy = $\sum_{i=1}^{c} - fi \log(fi)$ where fi is the frequency of the label I at a node and C is the number of unique labels.

Equation 3 Entropy Change Calculation

Information Gain is the splitting of data using entropy, it is the decrease in entropy after the dataset has been split.

Gain (T, X) = Entropy (T) – Entropy (T, X)

Equation 4 Information Gain formular

Where T= target variable

X= Features to be split on

Entropy (T,X) = The entropy calculated after the data split.

Figure 5.0 below shows the decision tree for the dataset used above.

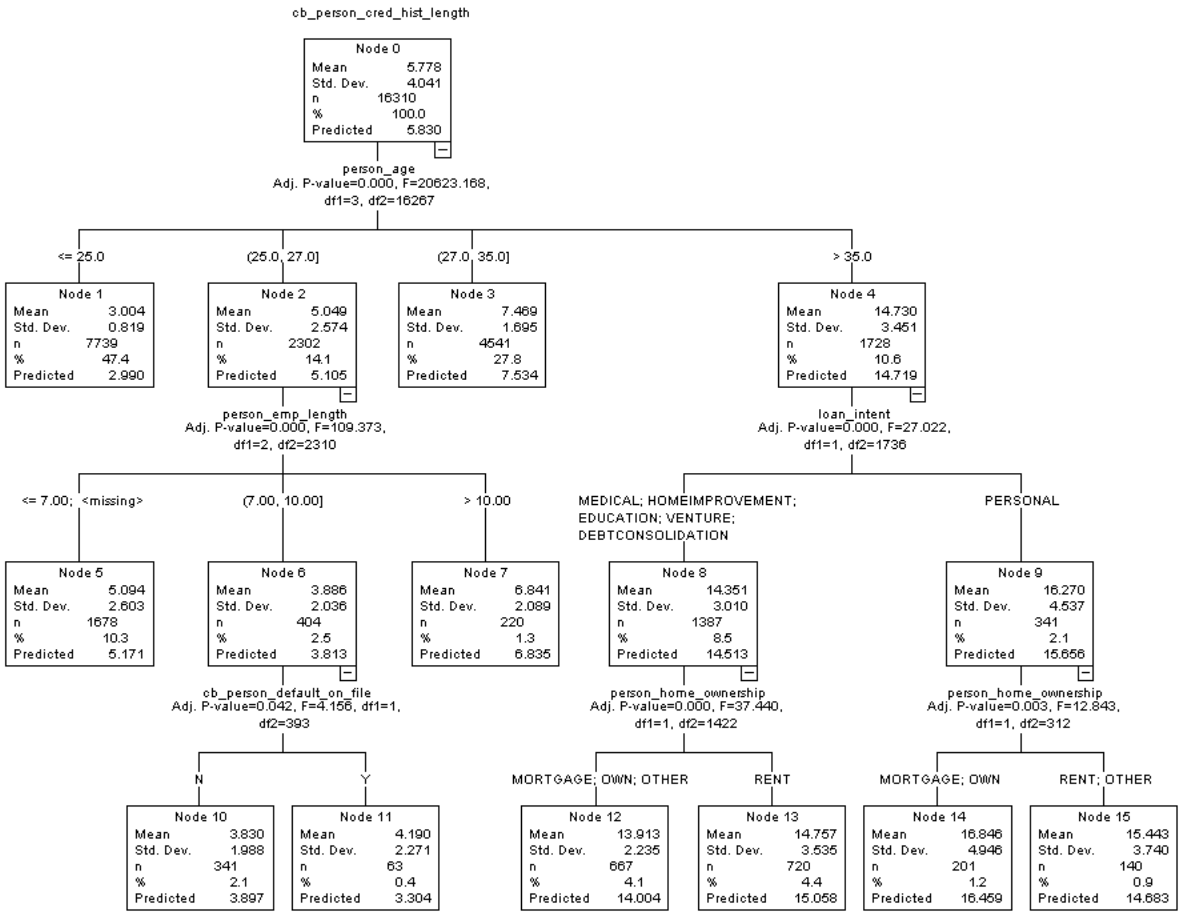

cb_person_cred_hist_length
Node 0 Mean 5.778 Std. Dev. 4.041 n 16310 % 100.0 Predicted 5.830
person_age Adj. P-value=0.000, F=20623.168, df1=3, df2=16267
<= 25.0
(25.0, 27.0]
(27.0, 35.0]
> 35.0
Node 1 Mean 3.004 Std. Dev. 0.819 n 7739 % 47.4 Predicted 2.990
Node 2 Mean 5.049 Std. Dev. 2.574 n 2302 % 14.1 Predicted 5.105
Node 3 Mean 7.469 Std. Dev. 1.695 n 4541 % 27.8 Predicted 7.534
Node 4 Mean 14.730 Std. Dev. 3.451 n 1728 % 10.6 Predicted 14.719
person_emp_length Adj. P-value=0.000, F=109.373, df1=2, df2=2310
loan_intent Adj. P-value=0.000, F=27.022, df1=1, df2=1736
<= 7.00; <missing>
(7.00, 10.00]
> 10.00
MEDICAL; HOMEIMPROVEMENT; EDUCATION; VENTURE; DEBTCONSOLIDATION
PERSONAL
Node 5 Mean 5.094 Std. Dev. 2.603 n 1678 % 10.3 Predicted 5.171
Node 6 Mean 3.886 Std. Dev. 2.036 n 404 % 2.5 Predicted 3.813
Node 7 Mean 6.841 Std. Dev. 2.089 n 220 % 1.3 Predicted 6.835
Node 8 Mean 14.351 Std. Dev. 3.010 n 1387 % 8.5 Predicted 14.513
Node 9 Mean 16.270 Std. Dev. 4.537 n 341 % 2.1 Predicted 15.656
cb_person_default_on_file Adj. P-value=0.042, F=4.156, df1=1, df2=393
person_home_ownership Adj. P-value=0.000, F=37.440, df1=1, df2=1422
person_home_ownership Adj. P-value=0.003, F=12.843, df1=1, df2=312
N
Y
MORTGAGE; OWN; OTHER
RENT
MORTGAGE; OWN
RENT; OTHER
Node 10 Mean 3.830 Std. Dev. 1.988 n 341 % 2.1 Predicted 3.897
Node 11 Mean 4.190 Std. Dev. 2.271 n 63 % 0.4 Predicted 3.304
Node 12 Mean 13.913 Std. Dev. 2.235 n 667 % 4.1 Predicted 14.004
Node 13 Mean 14.757 Std. Dev. 3.535 n 720 % 4.4 Predicted 15.058
Node 14 Mean 16.846 Std. Dev. 4.946 n 201 % 1.2 Predicted 16.459
Node 15 Mean 15.443 Std. Dev. 3.740 n 140 % 0.9 Predicted 14.683

Figure 5 Decision Tree Nodes Diagram.

3.2.5.6 Random Forest

This is a supervised machine-learning algorithm that can be used for both classification and regression. The random forest creates multiple decision trees as training. The decisions from multiple trees are used for its final predictions. It solves many complicated problems by integrating different learning principles. The trees are subdivided into several decision trees of different databases, and it's used to increase the prediction of data accuracy. It is a binary classification problem where a target variable is determined for prediction. It splits the decision tree using the decision nodes, by finding the best split by maximizing entropy gain. If it certifies a certain condition, the tree moves to the left, hence it moves to the right and finally reaches a leaf node where a class label is assigned to it.

They are highly sensitive to the training dataset. The process of creating a new dataset from a repetitive process is called bootstrapping. The process of combining results from multiple models is called aggregation. Bagging is a combination of bootstrap and aggregation. Fig 6.0 show the random forest technique for analyzing the dataset.

The node importance is calculated using Gini, by calculating the feature importance of each decision tree.

$$RFfi_i = \frac{\sum_j normfi_{i_j}}{\sum_{j \in allfeatures, k \in alltrees} normfi_{jk}}$$

Equation 5 Decision Tree

where RFfi (i)=the importance of the feature I calculated from all trees in the Random Forest model

Normfi(ij) = the normalized feature importance for I in tree j

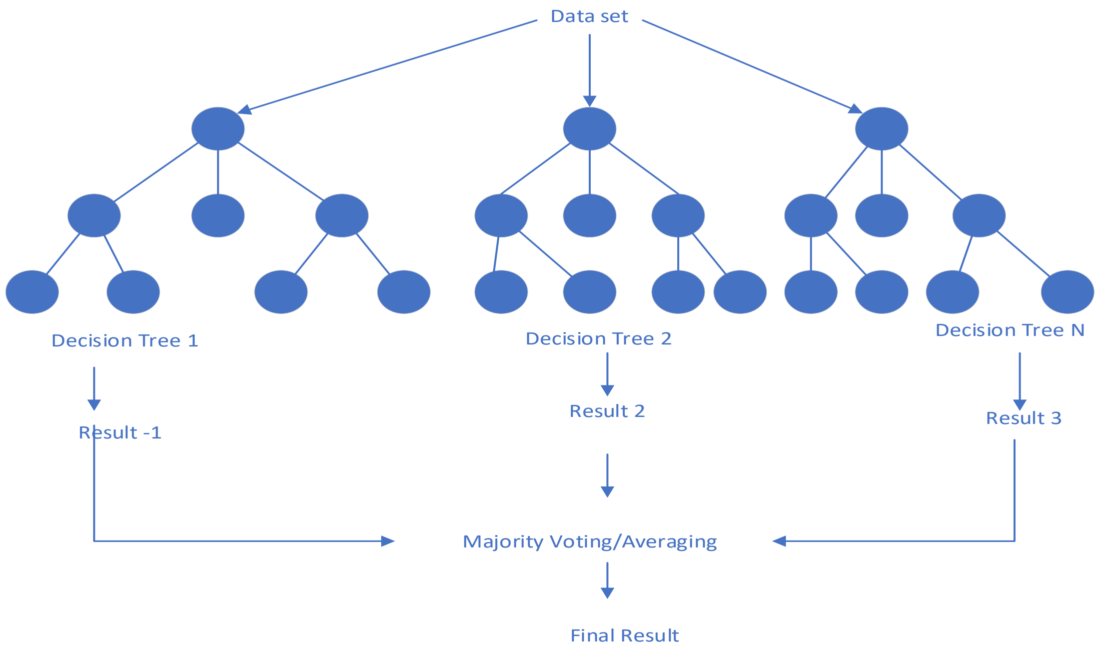


Figure 6 Random Forest Tree (Armel & Zaidouni, 2019)

3.2.5.7 The Naïve Bayes algorithm.

This is a classification algorithm which gives an assumption of independence among multiple predictors. It shows that the presence of a single feature is independent of each other. Each of the factors contributes to the probability independently. It is a good machine learning algorithm to use for large data sets. The algorithm uses Bayes theorem to train the classifier, which is called the probabilistic classifier. This calculates the probability distribution over a group of classes. The algorithm is naïve because it assumes all predictor variables are independent. The algorithm usually requires a small amount of training data to predict the expected results and are very fast.

$$p(y / X) = \frac{p(X / y).p(y)}{p(X)}$$ where p(x) & p(y) = probability events

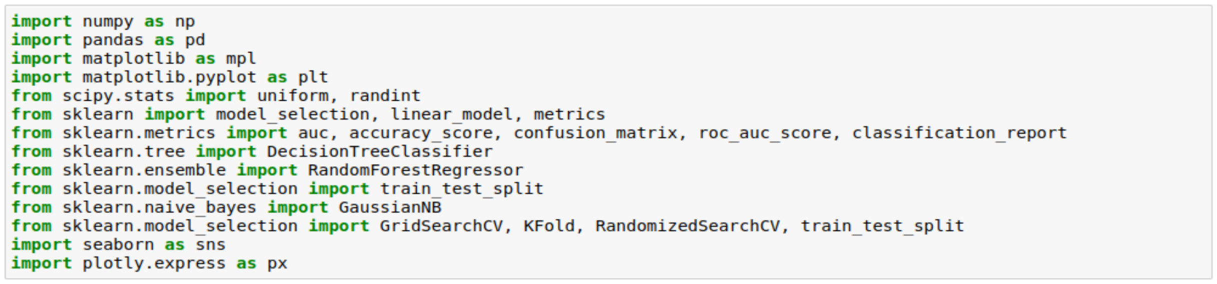
$$p(C_i \mid x_1, x_2, ..., x_n) = \frac{P(x_1, x_2, ..., x_n \mid c_i).P(C_i)}{P(x_1, x_2 ..., x_n)} \quad for 1 < i < k$$

Equation 6 Naive Bayes Equation

3.3 Result

The experiment was conducted using VMware pro-16, on a Ubuntu OS using Hadoop HDFS for the storage in the database using Apache Spark 3.3.1 and Jupyter notebook as the editor for execution. The Experiment analyzed the prediction of loan default using a supervised machine learning classifiers parameters such as Decision Tree, Random Forest, and Naïve Bayes according to 7 performance metrics such as Accuracy, precision, Recall, Auc, Roc, F1-score, weighted avg, and confusion metrics. The dataset was divided into training and testing for modeling in ratio of 70%: 30%. The machine learning algorithm with the best result was finally suggested. Below is the list of the results. Defaulters were denoted with 1, while 0 is denoted for active repayment capacity (non-defaulters). The confusion matrix for each of the machine learning models was determined. The True Positive and True Negative are usually considered as the value of choice for determining the prediction module.

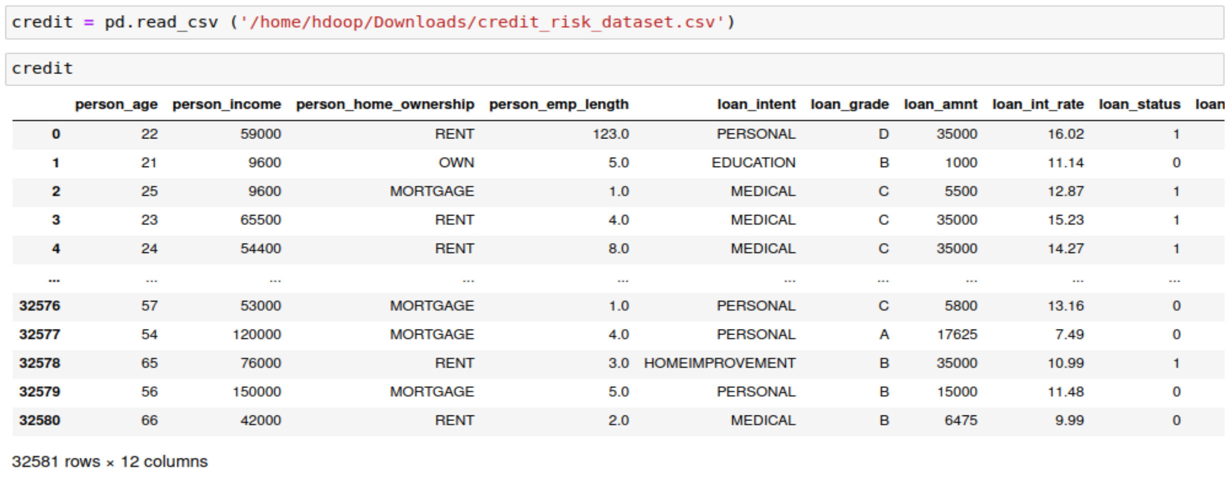
The following library was imported from pyspark for the machine learning algorithm. Figure 7.0 explains the different imported spark library for processing of the datasets.



```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from scipy.stats import uniform, randint
from sklearn import model_selection, linear_model, metrics
from sklearn.metrics import auc, accuracy_score, confusion_matrix, roc_auc_score, classification_report
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import GridSearchCV, KFold, RandomizedSearchCV, train_test_split
import seaborn as sns
import plotly.express as px
```

Figure 7 Importing Spark Libraries

The dataset used for the machine learning algorithm was Credit_risk_dataset.csv a Csv read from the Hadoop home environment. The file contains both categorical and numerical datasets which were first imported into the Jupiter notebook environment for proper analysis. The read data file showed 32,581 attributes and 12 columns. The summary of the total dataset imported is shown in figure 8.0.



```
credit = pd.read_csv ('/home/hdoop/Downloads/credit_risk_dataset.csv')
```

```
credit
```

	person_age	person_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_status	loan
0	22	59000	RENT	123.0	PERSONAL	D	35000	16.02	1	
1	21	9600	OWN	5.0	EDUCATION	B	1000	11.14	0	
2	25	9600	MORTGAGE	1.0	MEDICAL	C	5500	12.87	1	
3	23	65500	RENT	4.0	MEDICAL	C	35000	15.23	1	
4	24	54400	RENT	8.0	MEDICAL	C	35000	14.27	1	
...	...	...	...	...	...	...	...	...	...	
32576	57	53000	MORTGAGE	1.0	PERSONAL	C	5800	13.16	0	
32577	54	120000	MORTGAGE	4.0	PERSONAL	A	17625	7.49	0	
32578	65	76000	RENT	3.0	HOMEIMPROVEMENT	B	35000	10.99	1	
32579	56	150000	MORTGAGE	5.0	PERSONAL	B	15000	11.48	0	
32580	66	42000	RENT	2.0	MEDICAL	B	6475	9.99	0	

32581 rows × 12 columns

Figure 8 Credit Risk Dataset

3.3.1 Explanatory Data Analysis

Relationship between each of the categorical variables was gotten from the 8 Different conditions tested. I first tested the impact of changes in age versus loan status. The figure 9.0 below shows that the younger the age of the borrower the more the rate of default.

```
plt.figure(figsize=[20,10])
sns.countplot(x = 'person_age', hue= 'loan_status', data=credit);
```

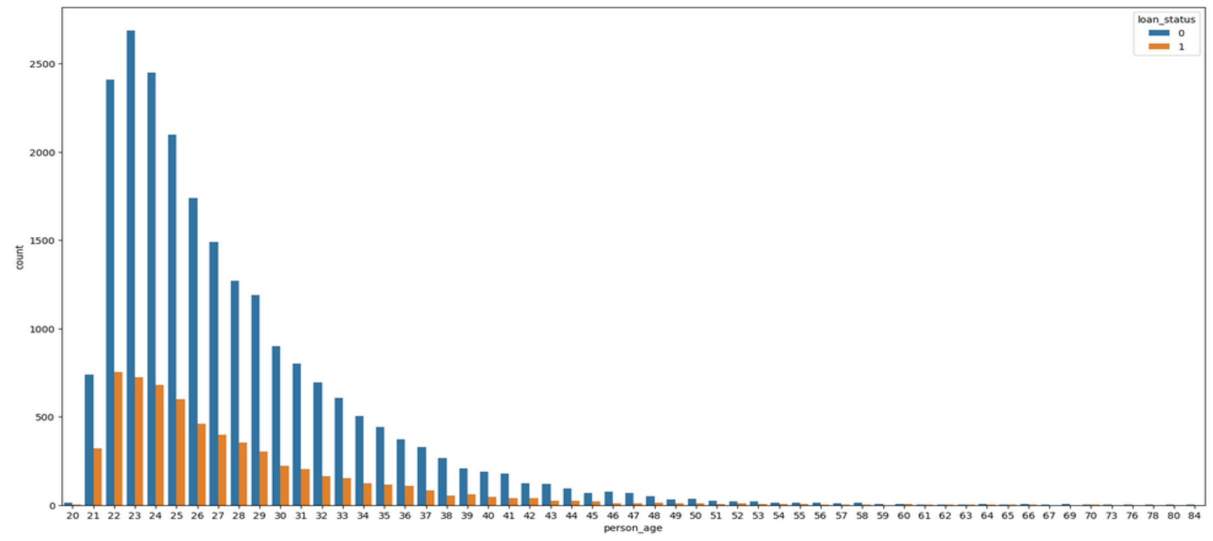


Figure 9 EDA Showing Relationship between Person Age and Loan Status

A visual relationship between each of the variables shows that the borrower has the capacity to default with a high rate of re-current expenditure. However, he has the capacity to repay when the money is invested on a business that can yield results such as a business venture or education. Figure 10.0 shows the relationship between loan intent and individual datasets.

```
#debtors shows more loan collected for medicals
fig1 = px.histogram(debtor,x='loan_intent',color='loan_intent',template='plotly_dark')
fig1.show()
```

```
# non debtors shows more loan for education
fig1 = px.histogram(no_debtor,x='loan_intent',color='loan_intent',template='plotly_dark')
fig1.show()
```

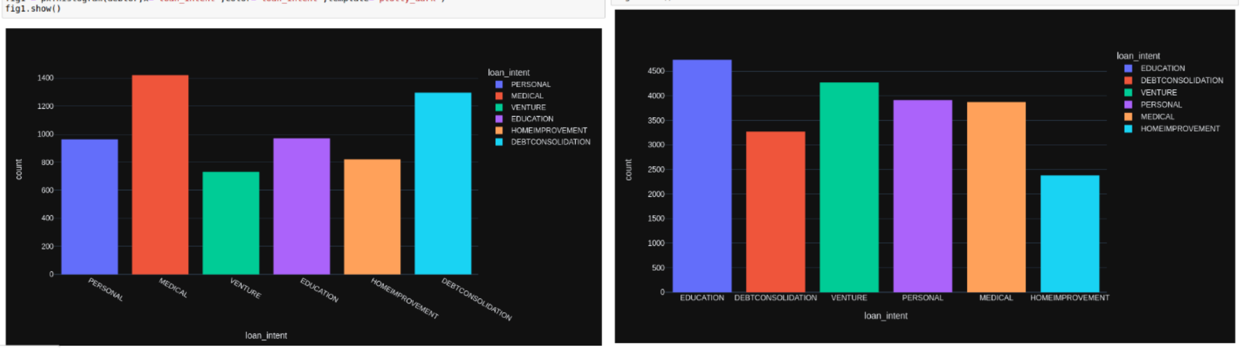


Figure 10 EDA Showing the Loan Intent

Below is a scatter plot diagram, fig 11.0 shows the relationship between personal age, income, loan amount. From the result above, there is a linear relationship between each of the variables. However, the changes in age largely affect the possibility of loan repayment.

```
#Scatter plot diagram
grafico = px.scatter_matrix(credit, dimensions=['person_age', 'person_income', 'loan_amnt'], color = 'loan_status')
grafico.show()
```

```
/home/hdoop/my_env/lib/python3.10/site-packages/plotly/express/_core.py:279: FutureWarning:

iteritems is deprecated and will be removed in a future version. Use .items instead.
```

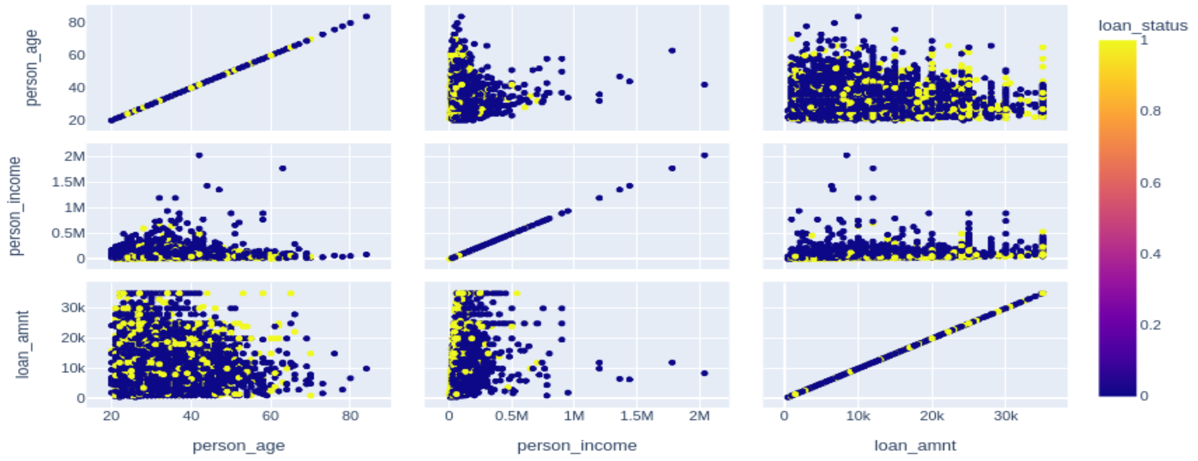


Figure 11EDA Showing the relationship between Person Age, income and amount on loan status.

3.3.2 Machine Learning Training

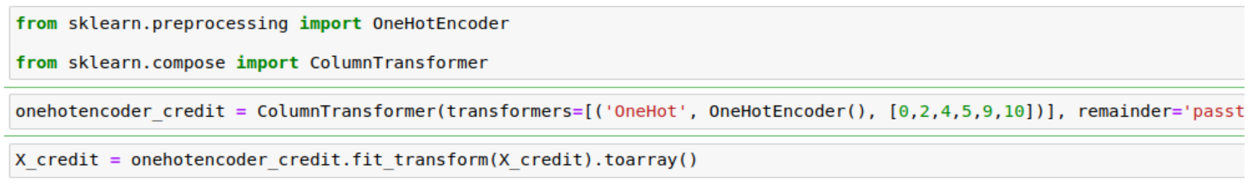
Machine learning algorithms were used to predict the probability of loan default, where the debtors had a loan status of 1 and the non-debtors were denoted with 0. Then the dataset was trained into a machine-readable format. Figure 12.0 shows the classification of the loan status trained into debtors and no debtor model.

```
debtor = credit[credit['loan_status']==1]
no_debtor= credit[credit['loan_status']==0]
```

```
# data processing
X_credit = credit.drop(columns= ['loan_status'])
X_credit
```

Figure 12 Label Classification into debtor and non-debtor

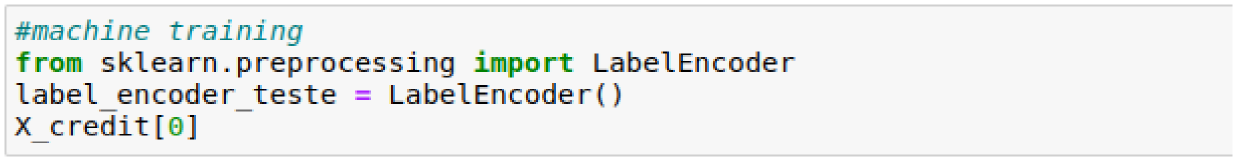
The dataset had both categorical and numerical datasets, hence the need to convert all categorical datasets to numerical ones, using OneHotEncoder from sklearn. preprocessing, giving it a numerical array. Fig 13.0 shows the machine learning library used to execute the task.



```
from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

onehotencoder_credit = ColumnTransformer(transformers=[('OneHot', OneHotEncoder(), [0,2,4,5,9,10])], remainder='passt

X_credit = onehotencoder_credit.fit_transform(X_credit).toarray()
```

Figure 13 Data Preprocessing using One-Hot Encoder

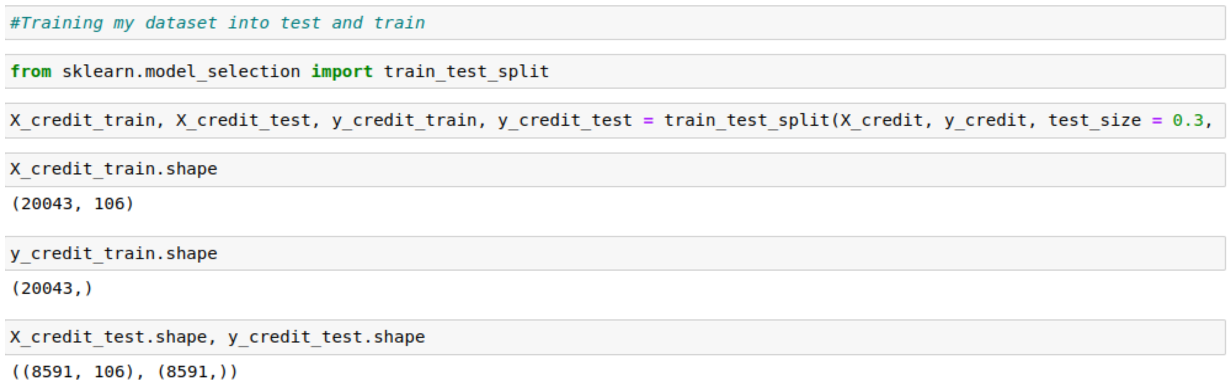
The label encoder converts labels into numerical form to make the dataset machine readable. Machine training was done using the sklearn preprocessing library as shown in figure 14.0.



```
#machine training
from sklearn.preprocessing import LabelEncoder
label_encoder_teste = LabelEncoder()
X_credit[0]
```

Figure 14 Class Label Encoder

The machine learning algorithm was trained using train_test_split from the sklearn.model_selection in order to enable the dataset to be understood by the machine to make an accurate prediction from the dataset supplied. The ratio of the train to test algorithm was 70:30, denoted with X_credit_train, X_credit_test, y_credit_test, and y_credit_test as illustrated below in figure 15.0:

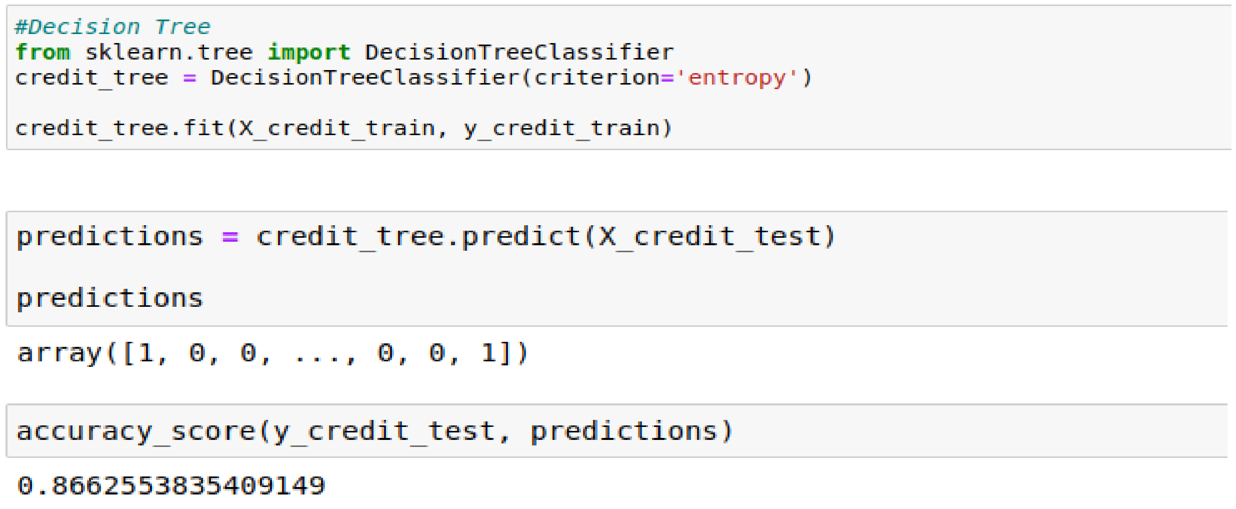


```
#Training my dataset into test and train

from sklearn.model_selection import train_test_split

X_credit_train, X_credit_test, y_credit_train, y_credit_test = train_test_split(X_credit, y_credit, test_size = 0.3,

X_credit_train.shape
(20043, 106)

y_credit_train.shape
(20043,)

X_credit_test.shape, y_credit_test.shape
((8591, 106), (8591,))
```

Figure 15 Preparing data for ML using Train-test-split.

3.3.2.1 Decision Tree method

The decision tree was first imported using sklearn.tree import DecisionTreeClassifier, then the dataset was trained to fit the test_train to create a prediction from the array of datasets. The illustration is shown in figure 16.0 below.



```
#Decision Tree
from sklearn.tree import DecisionTreeClassifier
credit_tree = DecisionTreeClassifier(criterion='entropy')

credit_tree.fit(X_credit_train, y_credit_train)
```

```
predictions = credit_tree.predict(X_credit_test)

predictions
```

```
array([1, 0, 0, ..., 0, 0, 1])
```

```
accuracy_score(y_credit_test, predictions)
```

```
0.8662553835409149
```

Figure 16 Decision Tree Classifier

The decision tree method gave a higher value of precision and recall which makes up the F1 Score. The decision tree gave a value for F1-Score of 91% for active repayment. The prediction accuracy was high with a value of 87%. Precision measures the probability that a value is correct, from the model below the data has a precision value of 92 %. The breakdown of the machine learning output for the decision tree algorithm is illustrated below in Table 2.0.

	Precision	Recall	F1-score	Support
0	0.92	0.91	0.91	6734
1	0.69	0.70	0.69	1857
Accuracy			0.87	8591

Table 2 Decision Tree Classification Report

The confusion matrix can be determined by selecting the True value for the matrix set. The True Positive and True Negative are usually considered as the value of choice. The decision tree predicted a higher value of TP to be 6134. i.e. using the decision tree method 6134 people that applied for a loan can repay while 1308 people would default using the confusion matrix model illustrated in figure 17.0 below. Table 3.0 is also used to show the output using a matrix of 0 and 1 scale.

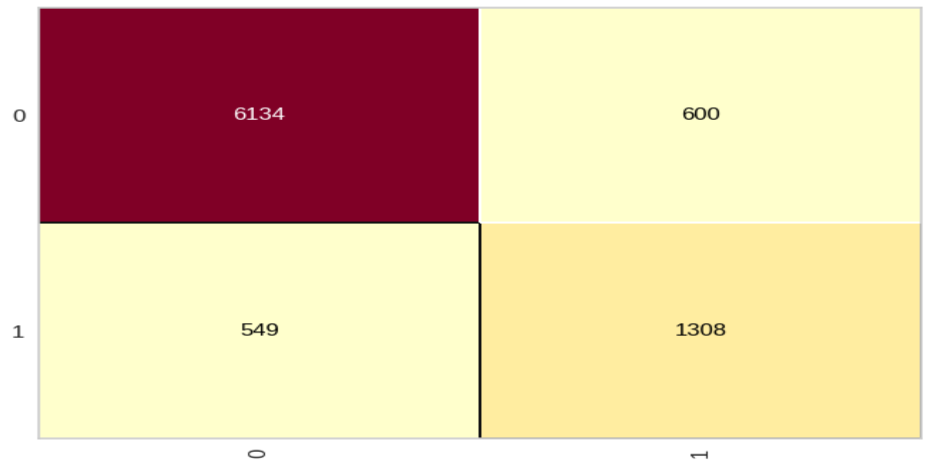


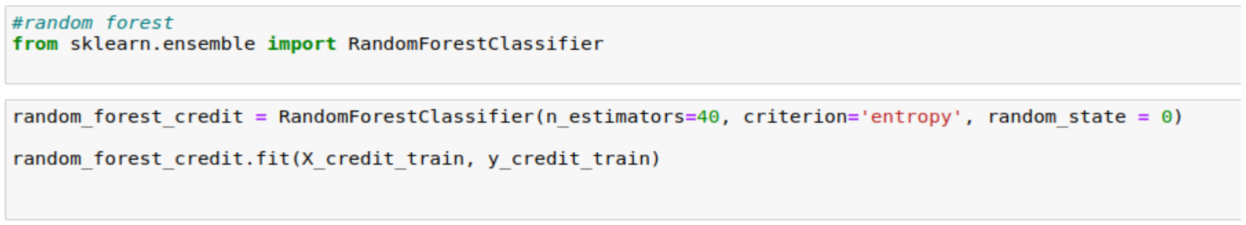
Figure 17 Confusion Matrix for Decision Tree Algorithms

Confusion Matrix	Predicted	
	0	1
0	6134	600
1	549	1308

Table 3 Summary of Confusion Matrix for Decision Tree

3.3.2.2 Random Forest

This is an ensemble for the machine learning algorithm, random forest is constructed from a large number of decision trees from the ensemble using the training dataset. The code for training the dataset using the machine learning test_train_split is illustrated in figure 18.0 below.



```
#random forest
from sklearn.ensemble import RandomForestClassifier
```

```
random_forest_credit = RandomForestClassifier(n_estimators=40, criterion='entropy', random_state = 0)

random_forest_credit.fit(X_credit_train, y_credit_train)
```

Figure 18 Random Forest Classifier

In the F1-Score, the ratio of the precision to Recall is significantly high with a value of 94% and a higher accuracy of 90%, it has a precision value of 91% and a recall value of 97%. It can be seen from the performance matrix parameters tested that the Random Forest method gave the highest accuracy of 90% as against other tested variables. This model has a good tendency to predict accurately based on the result gotten in Table 4.0 below.

	Precision	Recall	F1-score	Support
0	0.91	0.97	0.94	6734
1	0.86	0.66	0.74	1857
Accuracy			0.90	8591

Table 4 Summary Table of Confusion Matrix for Random Forest

The confusion Matrix below shows that the tendency for loan repayment is true with a positive accuracy value of 6537 and the tendency of default to be 1217 people from the dataset selected. The diagonal value performs the best analysis for the model according to the figure 19.0 shown below. Table 5.0 also shows the confusion matrix result after training the data set into binary output.

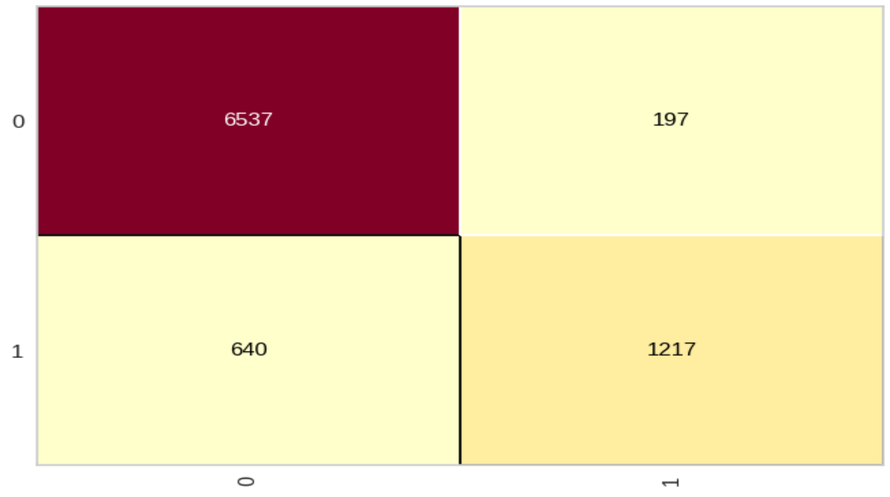


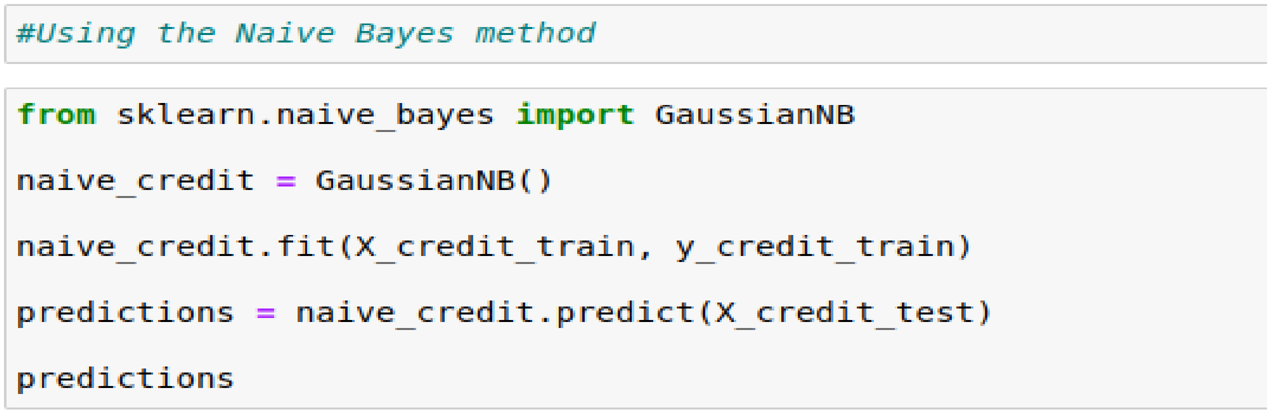
Figure 19 Confusion Matrix for Random Forest Classifier

Confusion Matrix	Predicted	
	0	1
0	6537	197
1	640	1217

Table 5 Summary table for Random Forest Classifier

3.3.2.3 Naïve Bayes

Naïve Bayes machine learning algorithm was imported from the python library to generate a prediction model, then the data set was trained to fit on the classifier in order to perform the predictions. Figure 20.0 illustrates the training of the data set using the Naïve Bayes machine learning algorithm.



```
#Using the Naive Bayes method
```

```
from sklearn.naive_bayes import GaussianNB

naive_credit = GaussianNB()

naive_credit.fit(X_credit_train, y_credit_train)

predictions = naive_credit.predict(X_credit_test)

predictions
```

Figure 20 Naive Bayes Classifiers

The Naïve Bayes method was used to test for 5 performance matrices. The accuracy is the number predicted by the number of total predictions. The result for Naïve Bayes shows that the accuracy value is low at 22%. The F1-Score, which is the ratio of precision to recall shows a medium F1-Score of 0.02. Table 6.0 illustrates the result from the Naïve Bayes algorithm used above.

Prediction	Precision	Recall	F1-Score	Support
0	0.86	0.01	0.02	6734
1	0.22	0.99	0.36	1857
Accuracy			0.22	8591

Table 6 Summary table for confusion matrix

The confusion matrix is the amount of accuracy of the prediction. It has a predictive analysis of TP, (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). The figures below predict the loan repayment capacity to be 70 (True Positive), and the capacity of defaults to be 1846 (True Negative). The result from the confusion matrix is given in figure 20.0 and table 7.0 below.

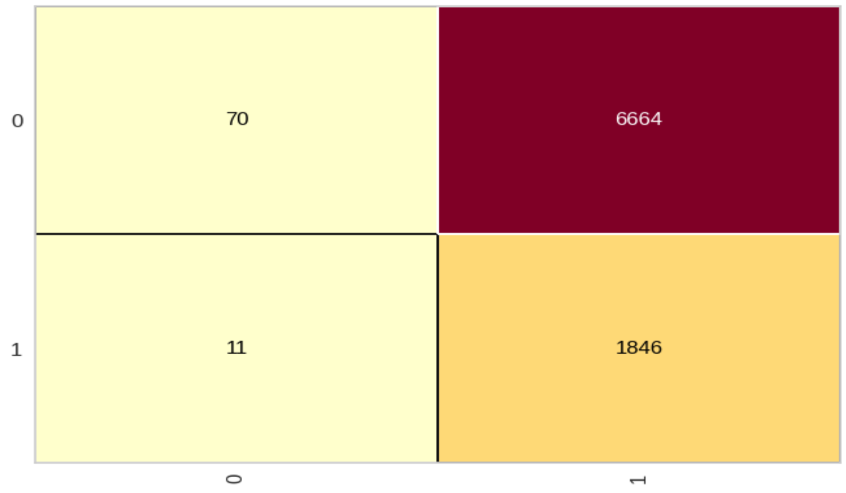


Figure 21 Confusion Matrix for Naive Bayes

Confusion Matrix	Predicted	
	0	1
	70	6664
	11	1846

Table 7 Summary Table of confusion matrix for Naive Bayes

3.3.3 Discussion

In the conducted experiment, a comparison of machine learning performance metrics for three classifiers was performed. The results indicate that the Random Forest ensemble machine learning technique outperformed the other models. Specifically, Random Forest achieved an accuracy of 90%, an F1 score of 94%, a precision of 91%, and a recall of 97%, surpassing the results obtained by the other machine learning algorithms. This suggests that ensemble machine learning techniques, as demonstrated in this study, offer superior performance in this context.

3.3.3.1 Summary of the Machine Learning Processes

The results for the three machine learning algorithms used are presented in Table 8.0, with the performance metrics divided into precision, recall, F1-score, and accuracy for analysis.

Machine Learning Results	Decision Tree	Random Forest	Naïve Bayes

Precision	92%	91%	86%
Recall	91%	97%	1%
F1-Score	91%	94%	2%
Accuracy	87%	90%	22%

Table 8 Summary Table of the Machine Learning Algorithms used.

3.3.4 Conclusion

Credit risk is better mitigated and managed by understanding the nature of borrowers' spending habits. Credit risk management is shown to be effectively managed by using big data through machine learning using spark than normal traditional processes. From the above result, Radom Forest gave the best output among other machine learning algorithm tested. The accuracy, precision, F1 score, and other performance index used suggested Random Forest as the most effective machine learning algorithm in predicting the probability of default and hence managing risk more effectively. However, the result shows that the accuracy of the dataset could vary based on the machine learning techniques used. Therefore, there is no best machine learning algorithm for making accurate prediction, the best result based on the data set used is always generally accepted to be the best model.

CHAPTER 4

4 STUDY 2- EMAIL CLASSIFICATION OF TEXT DATA USING MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING TECHNIQUE

4.1 Introduction

Increase in the number of internet users has significantly made email communication the most extensive use for individuals and businesses, however this has led to the emergence of unsolicited emails and information leakage caused by spam and phishing emails (Mujtaba et al., 2017) . Phishing emails are considered more dangerous because they target sensitive information from their users such as usernames, passwords, card numbers or pins, from unsuspecting employees or individuals. Meanwhile spam messages could lead to burden on email users due to their volume and frequency, high network bandwidth, large memory space and sometimes malware product attachments. The global average total cost of a data breach was $4.35 million in 2022, were stolen and compromised information are responsible for 19% of breaches and phishing was responsible for 16% (Bacanin et al., 2022).Different research has been developed for the classification of spam emails systems using supervised machine learning techniques, the objective of this research is to solve class imbalance problems, develop a novel spam-ham-phishing dataset and use the principles of machine learning and natural language processing to improve the performance of the three class of dataset developed namely (spam-ham, ham-phishing, spam-phishing) . To avoid bias due to the imbalanced dataset SMOTE (Synthetic over-sampling technique) was used to over-sample the minority, various data pre-processing method was performed on the dataset such as Tokenization using the word and sentence tokenizer for word count analysis, removal of stop words, stemming using the porter stemmer and lemmatization as the data cleaning process. Then different features extraction and selection was performed before training of the dataset using 5 different machine learning algorithms. The dataset used to train the model consists of 70% of the entire data, while 30% was used to test the model accuracy and performance. The result of the 5-machine learning (SVM, XGBoost, Random Forest, Multinominal and Gaussian Naïve bayes) shows how SMOTE improved the performance index of the classification techniques used. Consequently, the impact of solving the imbalance dataset using the SMOTE technique greatly improved the performance of the result, it shows that XGBoost performs better using the three-dataset developed. The spam-ham dataset had an accuracy of 0.99, precision of 1.00, recall of 0.98, and f1-score of 0.99, the spam-phishing likewise has an accuracy of 0.95, precision of 0.94, recall of 0.95, f1-score of 0.95 and ham-phishing dataset having an accuracy of 0.99, precision of 1.00 recall of 0.98 and f1-score of 0.94. Hence the result shows that XGBoost machine learning algorithms outperformed other algorithms using the datasets.

4.2 Background

The system architecture for classification of spam, ham, and phishing using machine learning is designed below to categorize incoming messages efficiently and accurately, such as emails or text messages, into one of three classes: "spam" (unsolicited or unwanted messages), "ham" (legitimate and desired messages), or "phishing" (fraudulent messages attempting to deceive recipients). This architecture is essential for protecting users from unwanted or harmful content and ensuring the integrity of digital communications.

4.2.1 System Architecture

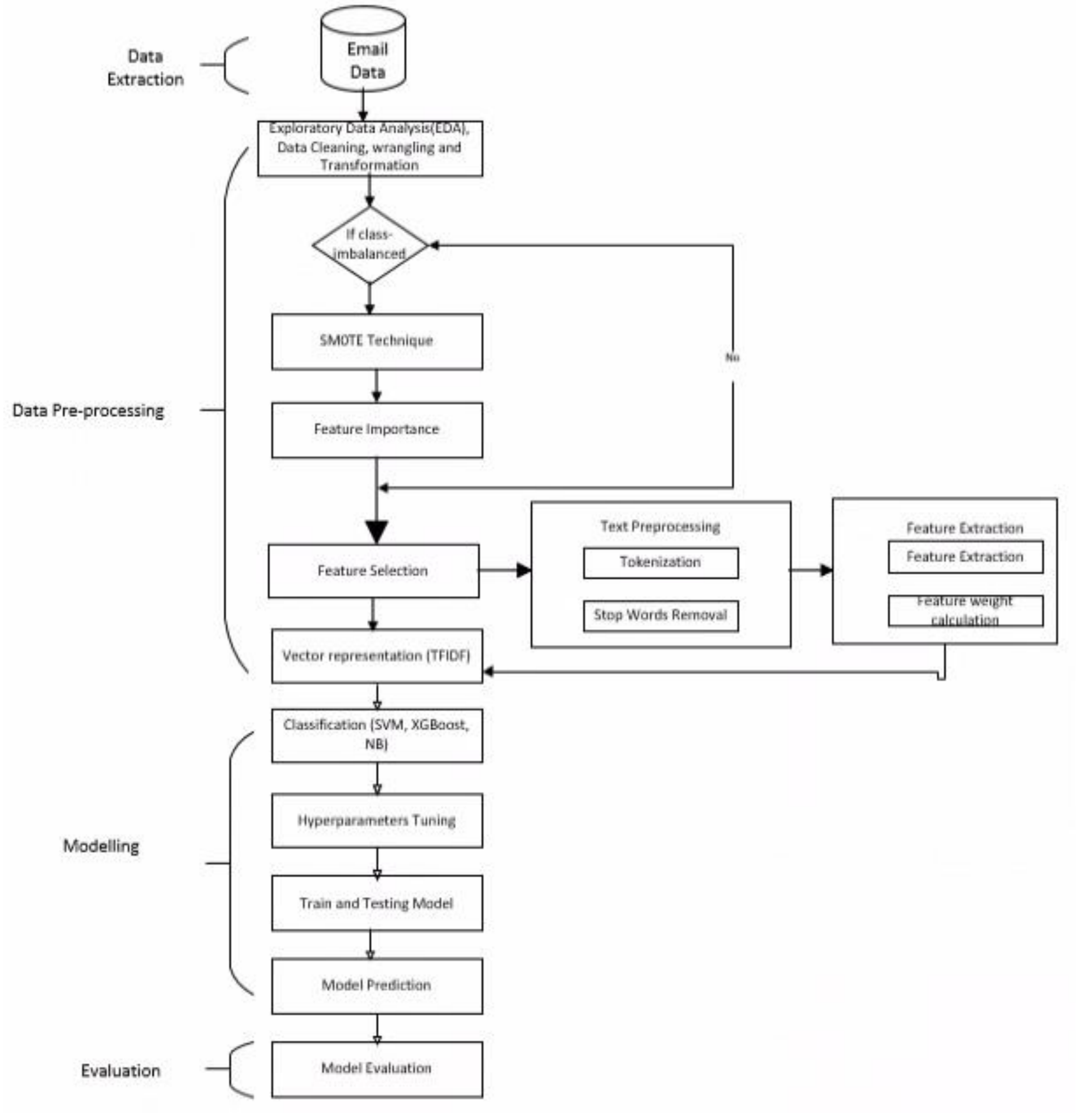


Figure 22 Proposed Methodology for Email Classification

4.2.2 Data Extraction

The dataset used comprises spam, ham, and phishing. The dataset was gotten from an educational institute phishing data, while the spam-ham dataset was gotten from Kaggle. Concatenation of the dataset was done to ensure data homogenization and integrity using Microsoft excel spreadsheet, converted into CSV file for proper processing and further transformed for machine learning classification.

4.2.3 Exploratory data analysis (EDA)

EDA was done to give better understanding and insights into the dataset, identify data quality issues, outlier detection and give an effective visualization to further gain insights.

4.2.4 SMOTE Technique

The term "Synthetic Minority Over-sampling Technique" refers to a method frequently used in machine learning to handle the problem of class imbalance within a dataset. This imbalance happens when the total number of instances in one class called the Minority class is much lower than the total number of instances in the other class. The dataset utilized was balanced by synthetic oversampling of the minority class.

4.2.5 Feature importance

This is a technique used to determine how important a feature variable is in predicting the target variable or output. It helps to identify features that have the most significant impact on the model's performance and prediction.

4.2.6 Feature selection

A key step in improving a model's performance in machine learning is feature selection. It entails selecting a subset of the available features (sometimes referred to as variables or characteristics) from the dataset carefully and eliminating unnecessary or redundant ones. The main objective is to increase the model's precision, effectiveness, and interpretability.

4.2.7 Text Preprocessing:

This involves cleaning and transforming raw text into a format suitable for a machine learning model. Examples of text preprocessing methods are Lowercasing, tokenization, removal of punctuation, stop words removal, stemming and lemmatization.

4.2.8 Vector representation:

Text is converted into numerical vectors through vector representation in NLP, which is essential for machine learning. TF-IDF and Bag-of-Words (BoW) are two popular techniques. BoW visualizes text as vectors, where each dimension indicates the frequency of a word within the text. Machine learning algorithms can process language effectively by turning text into vectors. These techniques are fundamental to NLP and enable machine language analysis and comprehension (Schütze et al., 2008).

4.2.9 Machine Learning Classification Technique

Classification techniques are machine learning frameworks designed to classify incoming emails into one of three categories: "spam," "ham" (legitimate), or "phishing." In this research the algorithm leverages the strengths of three prominent machine learning techniques: Support Vector Machines (SVM), XGBoost (Extreme Gradient Boosting), and Naive Bayes.

4.2.10 Train and Test Instance

In machine learning, the concept of train and test instances is crucial for evaluating model performance. The training set (70% of the dataset in this research) is used to train the model, enabling it to learn patterns and features from the data, such as distinguishing between spam, legitimate (ham), and phishing emails. The testing set (30% of the dataset) serves as a validation set, containing data the model has not seen during training. It assesses the model's ability to generalize and make accurate predictions on unseen instances, mirroring real-world scenarios. Performance metrics are then calculated using the testing set to gauge the model's effectiveness in classifying different types of emails. This division ensures robust and reliable model evaluation.

4.2.11 Hyperparameter tuning.

For improved generalization and accuracy, hyperparameter modification in spam-ham and phishing email categorization optimizes model parameters. The learning rate, regularization, and tree depth are important hyperparameters. Techniques like random and grid searches look at different parameter combinations. Achieving the best tuning improves spam detection while reducing false positives and negatives (James et al., 2013).

4.2.12 Model prediction

Model prediction is the stage of machine learning where a trained model makes predictions about the target variable (dependent variable or outcome) using new input data (independent variables). To get meaningful results, this procedure requires applying the discovered patterns and correlations from the training data to previously unexplored data. It's an important step when applying machine learning models to applications like email classification, where the model's capacity to correctly identify an email as spam, ham, or phishing is crucial for email security and user experience (Bishop & Nasrabadi, 2006) .

4.2.13 Model evaluation

Model evaluation is usually one of the final stages in the machine learning processes where the performance of unseen data is assessed on the unseen data. The model performance is typically measured using various evaluation metrics. In this research the classification report consisting of the accuracy, precision, recall, F1 score, and the confusion matrices was used to determine the performance index.

4.3 Methodology

The research methodology comprises five core stages. Firstly, data acquisition involved obtaining phishing data from an educational institute and ham-spam data from Kaggle. In the second stage, dataset pre-processing encompassed tasks like handling empty rows and columns, converting email text to lowercase, removing punctuation to eliminate special characters, excluding non-meaningful or non-English words through WordNet, lemmatization using WordNet lemmatizer, and eliminating stop words to emphasize crucial terms. Subsequently, the pre-processed data underwent the feature selection and extraction stage. The fourth step involved employing five machine learning classifiers, namely Support Vector Machine (SVM), XGBoost, Naïve Bayes (Multinomial and Gaussian), and Random Forest, on the chosen attributes. The entire experiment was conducted using Google Colab through the Jupyter notebook IDE. The outcomes and evaluation of the machine learning algorithm are detailed in the subsequent sections.

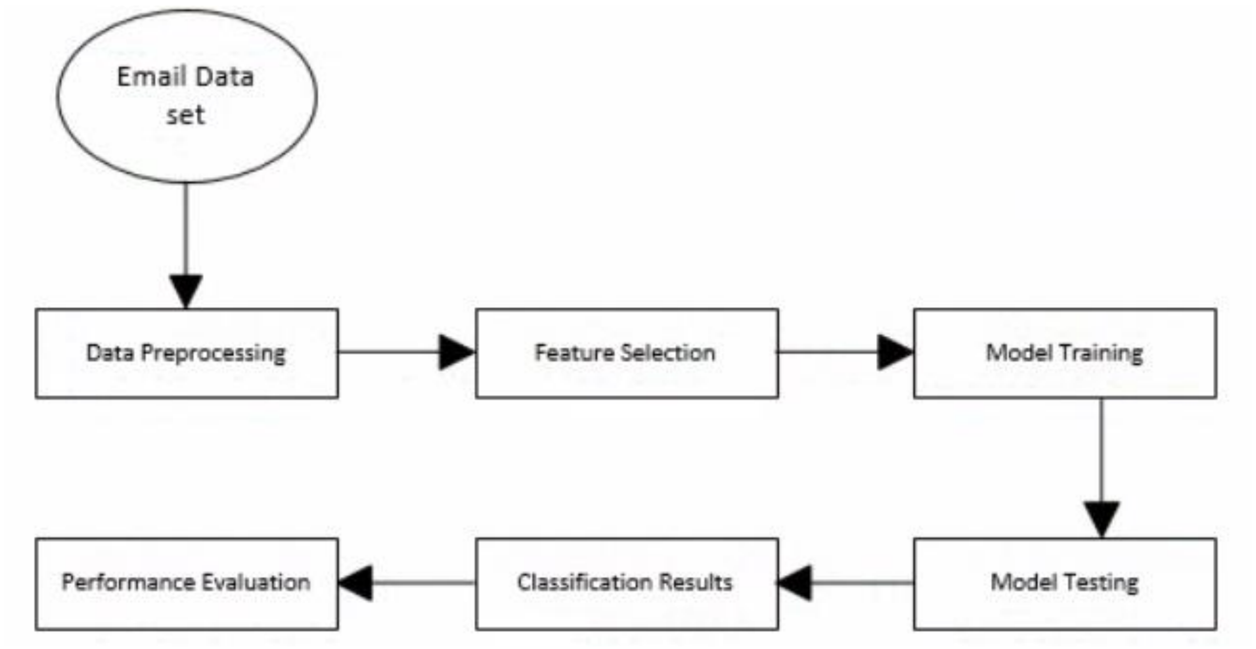


Figure 23 Text Classification Process (Kotsiantis et al., 2006)

4.3.1 Dataset

Getting the right dataset for effective spam-ham-phishing classification is of paramount importance. The dataset employed in this study originates from two distinct sources. Firstly, data was sourced from Tanusree Sharma's GitHub repository, encompassing educational institute phishing data (https://github.com/TanusreeSharma/phishingdataAnalysis/blob/master/1st%20data/PhishingEmailData.csv). Additionally, spam-ham email data was obtained from Kaggle (https://www.kaggle.com/datasets/shantanudhakadd/email-spam-detection-dataset-classification). To create the dataset, these sources were amalgamated, followed by cleaning and segmentation into three distinct categories: spam-ham, spam-phishing, and ham-phishing. Consequently, the dataset employed for this research comprises three primary classes: ham, spam, and phishing data. Specifically, the ham dataset encompasses 4,825 instances and comprises two attributes. It incorporates 747 spam emails and 189 phishing emails. The dataset features two columns: "V1" denoting the email class (spam, ham, or phishing) and "V2" representing the email body in supervised learning. In terms of class distribution, the spam dataset constitutes 12.9% of the entire dataset, while the ham class constitutes 83.7%. The phishing class, representing a more recent strategy employed by fraudsters, constitutes a mere 3.3% of the overall dataset. Further examination of the dataset reveals the following class combinations:

Spam-ham: 5,572 instances, accounting for 96.7% of the total dataset.

Spam-phishing: 936 instances, constituting 16.2%.

Ham-phishing: 5,014 instances, representing 87% of the total dataset.

4.3.1.2 Spam-Ham Dataset

Spam refers to unsolicited or unwanted messages sent over an electronic communication channel usually as emails. They could be sent as bulk messages containing advertisements, promotional content and sometimes malicious links. However, ham emails are legitimate and non-malicious, also known as non-spam. The dataset contains a collection of email messages, where the class is labeled as either "spam" or "ham". In this research the total number of the dataset is 5572, where 87.38% consist of the ham class with 4,825 instances and the spam class consist of 12.62% which is 747 instances of the dataset. The figure illustrates the ratio of the spam-ham dataset in a pie chart below.

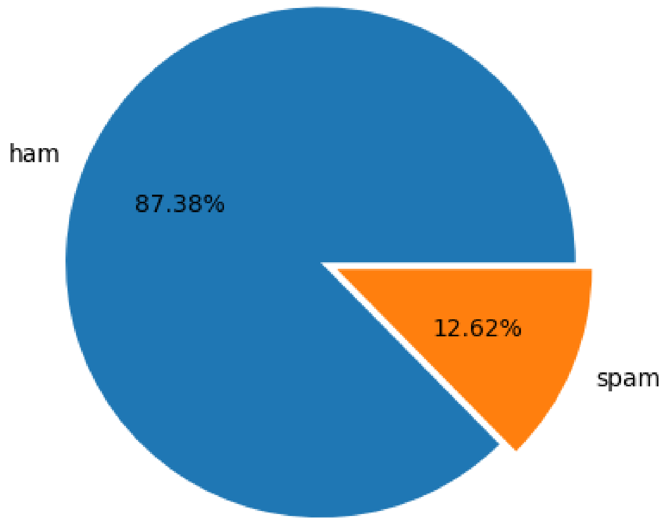
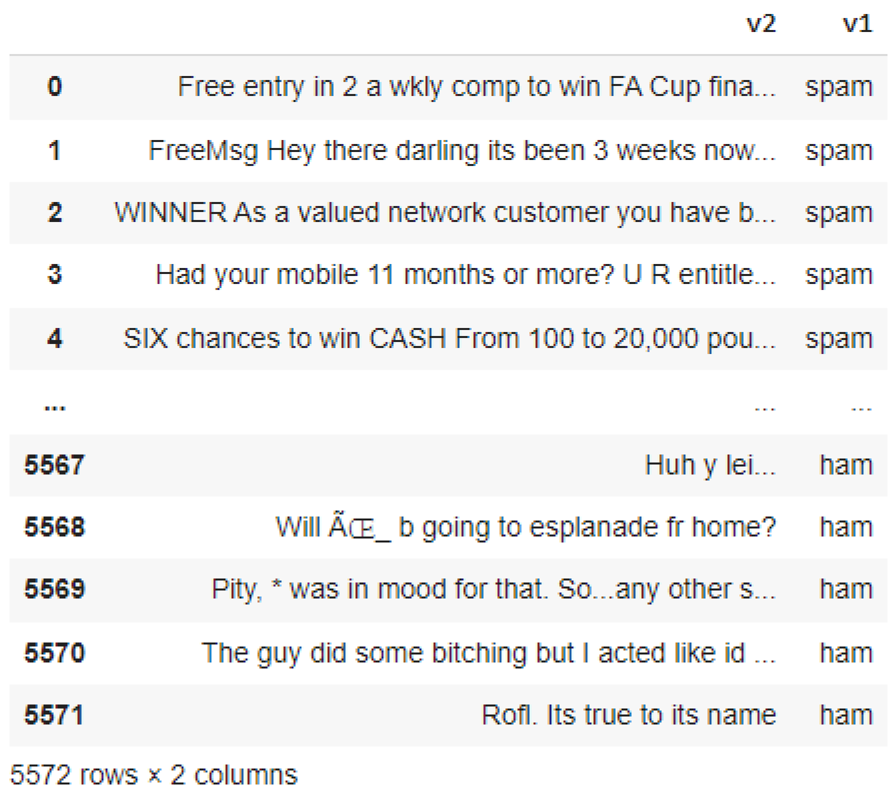


Figure 24 Pie Chart showing Spam-Ham Dataset

The main objective is to construct a machine learning model that will automatically categorize incoming messages as spam or ham-based dataset, depending on the model. The spam-ham dataset utilized for the machine learning model is shown in Table 9.0 below.



	v2	v1
0	Free entry in 2 a wkly comp to win FA Cup fina...	spam
1	FreeMsg Hey there darling its been 3 weeks now...	spam
2	WINNER As a valued network customer you have b...	spam
3	Had your mobile 11 months or more? U R entitle...	spam
4	SIX chances to win CASH From 100 to 20,000 pou...	spam
...	...	...
5567	Huh y lei...	ham
5568	Will ÃŒ_ b going to esplanade fr home?	ham
5569	Pity, * was in mood for that. So...any other s...	ham
5570	The guy did some bitching but I acted like id ...	ham
5571	Rofl. Its true to its name	ham

5572 rows × 2 columns

Table 9 Data Frame for Spam-Ham Dataset

4.3.1.3 Ham-phishing Dataset

Ham data encompasses genuine, non-malicious messages, typically regular emails that aren't spam or phishing; these are messages desired by the recipient. In contrast, phishing involves deceptive cyber-attacks orchestrated by fraudsters aiming to extract sensitive information like passwords, credit card numbers, or personal details. The ham-phishing dataset combines legitimate (ham) and phishing attack emails, serving as training and evaluation data for machine learning models designed to detect phishing emails. In this study, the ham dataset comprises 4,825 instances, while the phishing dataset includes 189 attributes. In the figure 25.0 below, the ham-phishing dataset is characterized by 96.25% ham and 3.75% phishing instances.

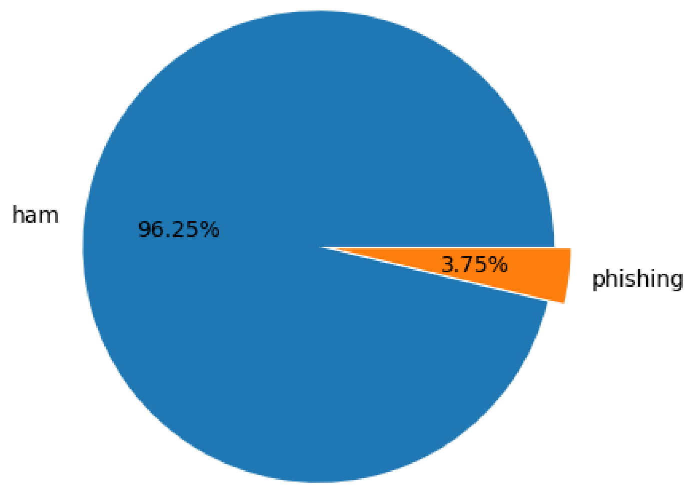
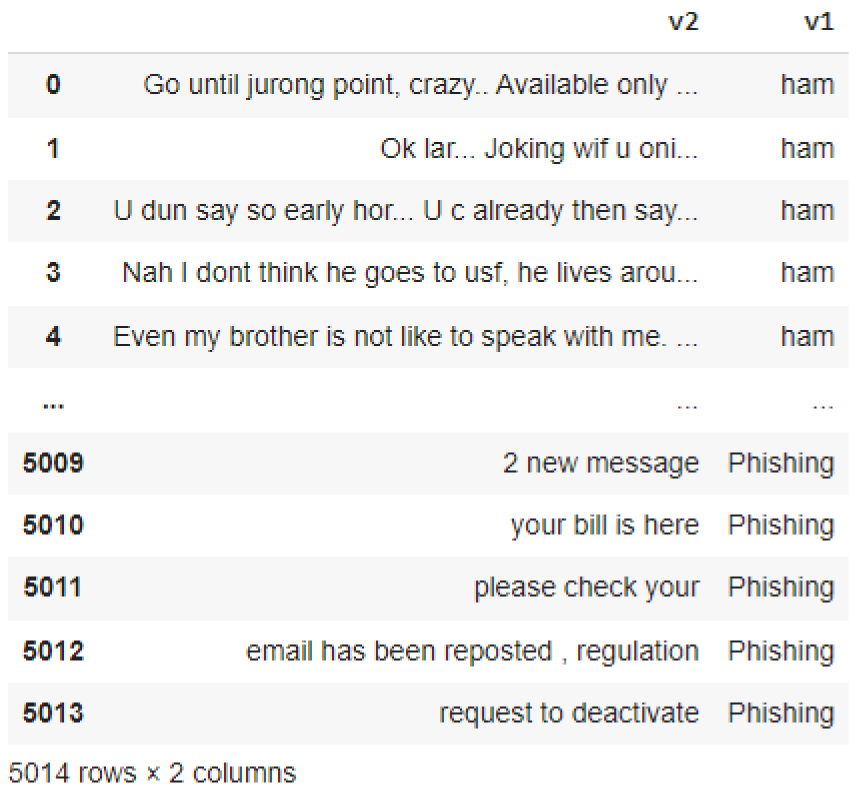


Figure 25 Pie Chart showing Ham-Phishing Dataset

Ham-phishing dataset is valuable for developing and testing machine learning models aimed at detecting phishing attacks in email communication. This would contribute significantly to cyber security measures and protect individuals and organizations from phishing attacks. Table 10 below illustrates the dataset used in the ham-phishing machine learning model.



	v2	v1
0	Go until jurong point, crazy.. Available only ...	ham
1	Ok lar... Joking wif u oni...	ham
2	U dun say so early hor... U c already then say...	ham
3	Nah I dont think he goes to usf, he lives arou...	ham
4	Even my brother is not like to speak with me. ...	ham
...	...	...
5009	2 new message	Phishing
5010	your bill is here	Phishing
5011	please check your	Phishing
5012	email has been reposted , regulation	Phishing
5013	request to deactivate	Phishing

5014 rows × 2 columns

Table 10 Data frame showing Ham-Phishing Dataset

4.3.1.4 Spam-Phishing Dataset

As previously mentioned, spam emails are unwanted, unsolicited, and often irrelevant messages inundating your inbox on the internet, typically attempting to entice you into purchasing products or services. Phishing emails, on the other hand, represent a deliberate fraudulent effort, where malicious actors target individuals or organizations with the primary goal of obtaining sensitive information. These emails often adopt the guise of legitimate entities to deceive recipients into taking actions that compromise security. The spam-phishing dataset amalgamates information from both spam and phishing emails. In this study, spam accounts for 78.74% of the dataset, while the remaining 21.26% is attributed to phishing instances. The accompanying figure 26.0, presented a pie chart, further explains the proportional distribution of spam-phishing data.

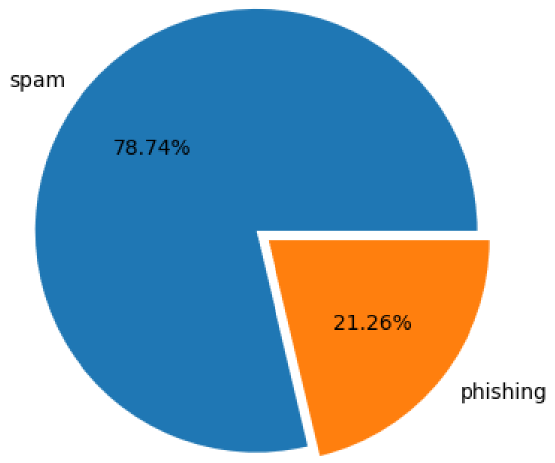
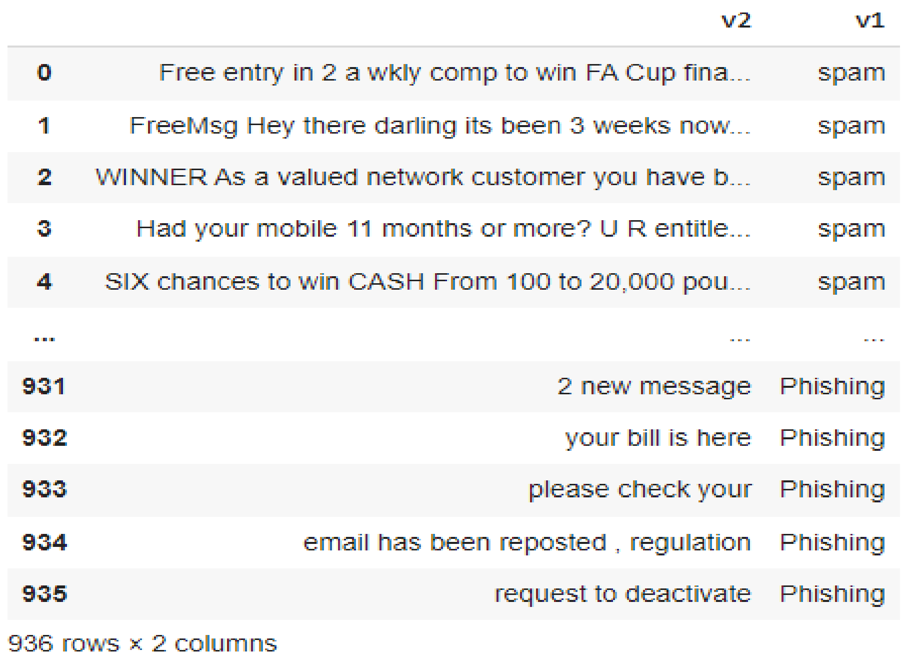


Figure 26 Pie Chart showing Spam-Phishing Dataset

Fig 26.0 shows the percentage of spam-phishing using a pie chart. The goal of the dataset is to develop and evaluate machine learning models that can detect both spam and phishing emails. The table below shows the email body and class of the dataset used for the machine learning model.



	v2	v1
0	Free entry in 2 a wkly comp to win FA Cup fina...	spam
1	FreeMsg Hey there darling its been 3 weeks now...	spam
2	WINNER As a valued network customer you have b...	spam
3	Had your mobile 11 months or more? U R entitle...	spam
4	SIX chances to win CASH From 100 to 20,000 pou...	spam
...	...	...
931	2 new message	Phishing
932	your bill is here	Phishing
933	please check your	Phishing
934	email has been reposted , regulation	Phishing
935	request to deactivate	Phishing

936 rows × 2 columns

Table 11 Data Frame showing Spam-Phishing Dataset

4.3.2 Exploratory data Analysis

Exploratory data analysis (EDA), which involves examining and understanding the dataset before modeling, is an essential part of machine learning and natural language processing (NLP). Two columns and 5,762 rows make up the dataset, where each row stands for a single data instance. EDA uses a variety of statistical and graphical approaches to extract insightful information from the dataset. Word frequency analysis, sentence length analysis, average word length analysis, and word cloud production are a few of these methods. These techniques give a deeper comprehension of the traits and patterns in the dataset. Another useful tool in EDA is the word cloud, a graphic representation of text data. With word size and color indicating their prominence, it visualizes the frequency and significance of terms in the dataset. EDA, in general, acts as a foundational step to get deeper understanding of the properties of the dataset, permitting deeper conclusions during later machine learning and NLP activities.

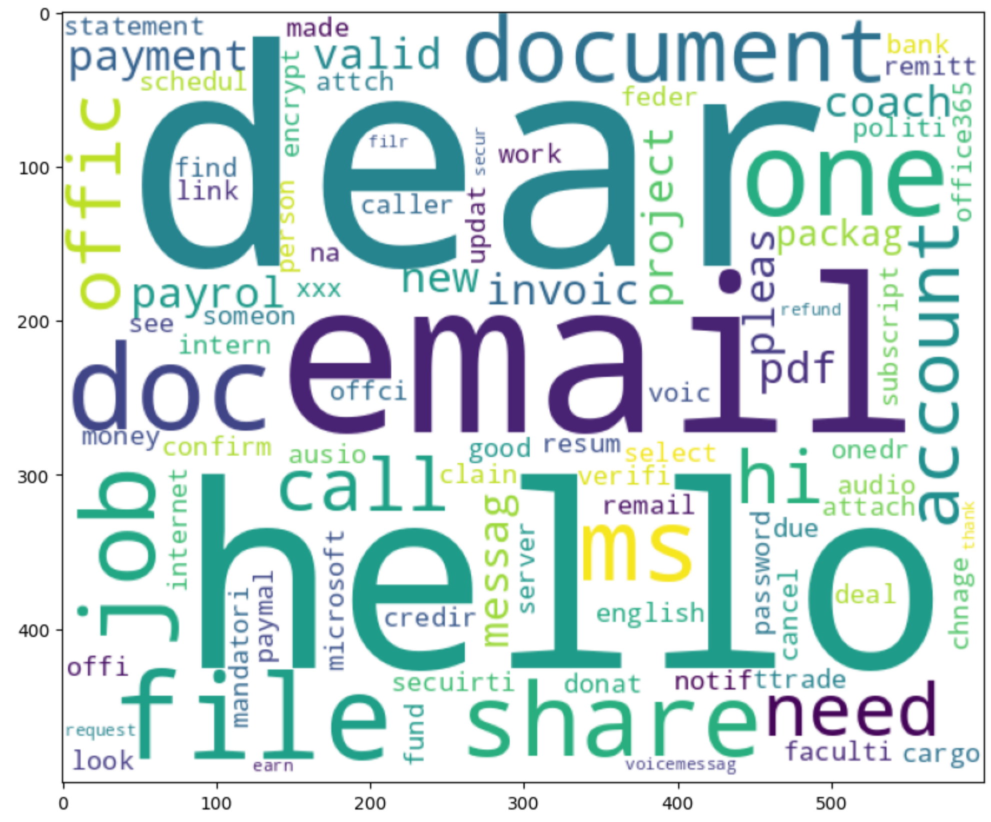


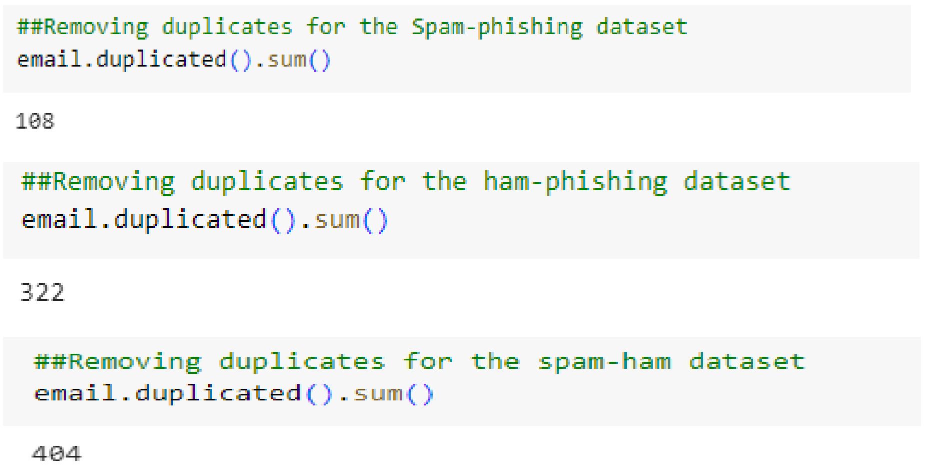
Figure 27 Word Cloud Visualization of the Dataset

4.3.3 Data extraction

The dataset used is a public text data from Kaggle consisting of spam-ham, and phishing dataset. It contains two columns namely email and label class. The dataset was downloaded from a Tanusree Sharma GitHub account consisting of an educational institute phishing data (https://github.com/TanusreeSharma/phishingdataAnalysis/blob/master/1st%20data/PhishingEmailData.csv). while the spam-ham email was gotten from Kaggle (https://www.kaggle.com/datasets/shantanudhakadd/email-spam-detection-dataset-classification). The dataset was loaded using google Collaboratory on a jupyter notebook and different python libraries were used to perform the analysis.

4.3.4 Missing/ Duplicate values

The dataset used for the machine learning classification had some unnamed columns consisting of unnamed 2 and unnamed 3, with NaN data available in the rows. The dataset with NaN rows and columns was first removed from the table as part of the data-cleaning process, then the number of duplicate values was counted. Spam-ham dataset consisted of 404 duplicates, spam-phishing consisted of 108 duplicates and ham-phishing consisted of 322 duplicates. To avoid a biased performance of the model, the duplicate entries were removed from the dataset. The figures below show the number of duplicates removed from the dataset during the data preprocessing.



```
##Removing duplicates for the Spam-phishing dataset
email.duplicated().sum()
```

```
108
```

```
##Removing duplicates for the ham-phishing dataset
email.duplicated().sum()
```

```
322
```

```
##Removing duplicates for the spam-ham dataset
email.duplicated().sum()
```

```
404
```

Figure 28 Data Cleaning Process of the entire dataset

4.3.5 Data preprocessing

Data extracted from the real world are usually inadequate, consisting of noise and missing values. Hence, the need for data pre-processing is imperative to transform them from a dirty or incomplete stage into a clean, usable, and organized form. Inconsistencies in a dataset can include typos, missing data, and data with different scales. The dataset was first prepared for use in the model. Examples of data pre-processing done on the email dataset includes Text cleaning, Conversation to lowercase, Tokenization, Removal of stop words, stemming/ lemmatization, and features extraction. Skipping this important stage in a machine learning model would affect the result because most models can't handle missing values, while some are affected by outliers, high dimensionality and noisy data preprocessing makes the dataset completer and more accurate.

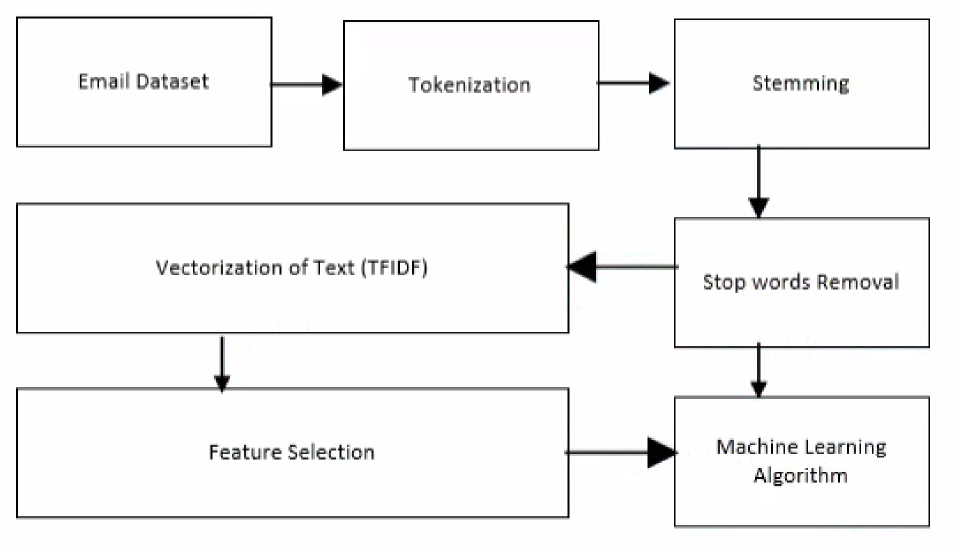
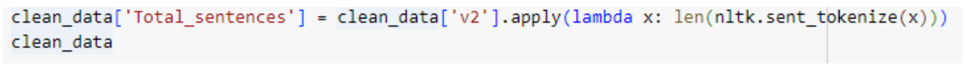


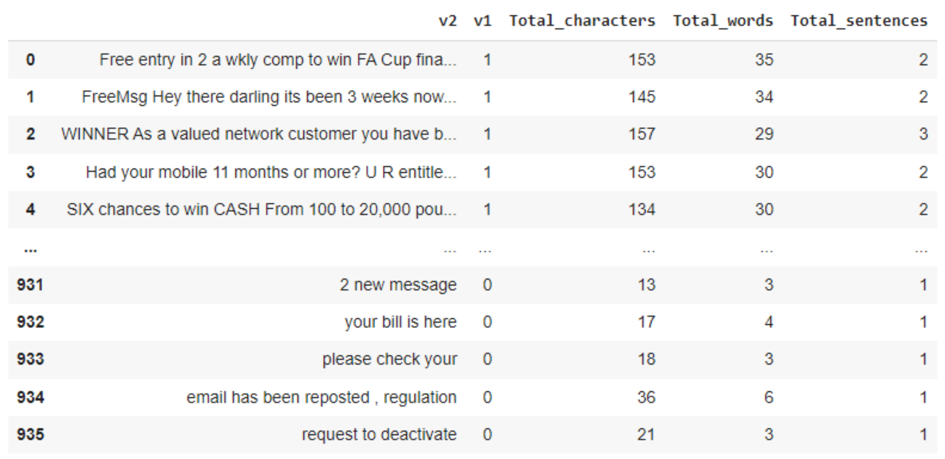
Figure 29 Data Preprocessing workflow

4.3.5.1 Tokenization

Tokenization plays a pivotal role in Natural Language Processing (NLP), being indispensable for effectively handling text data. It involves breaking down lengthy text into smaller units referred to as tokens, as explained by Pai (2020). These tokens can encompass words, characters, or sub-words (n-gram characters), depending on the specific tokenization method employed. Among these methods, Word Tokenization stands out as a widely utilized algorithm, where text is segmented into individual words. This approach is exemplified by pre-trained word embeddings such as Word2Vec and GloVe. Conversely, Character Tokenization dissects text into sets of individual characters, adept at handling Out of Vocabulary (OOV) terms while preserving word information. Sub-word Tokenization, on the other hand, disassembles text into sub-word components or N-gram characters, separated by spaces or punctuation characters (referred to as delimiters). Traditional NLP techniques like Count Vectorizer and TF-IDF leverage vocabulary as features, enhancing model performance. In this study, word tokenization was adopted to segment the text data into tokens, followed by the calculation of the total word count for subsequent analysis.

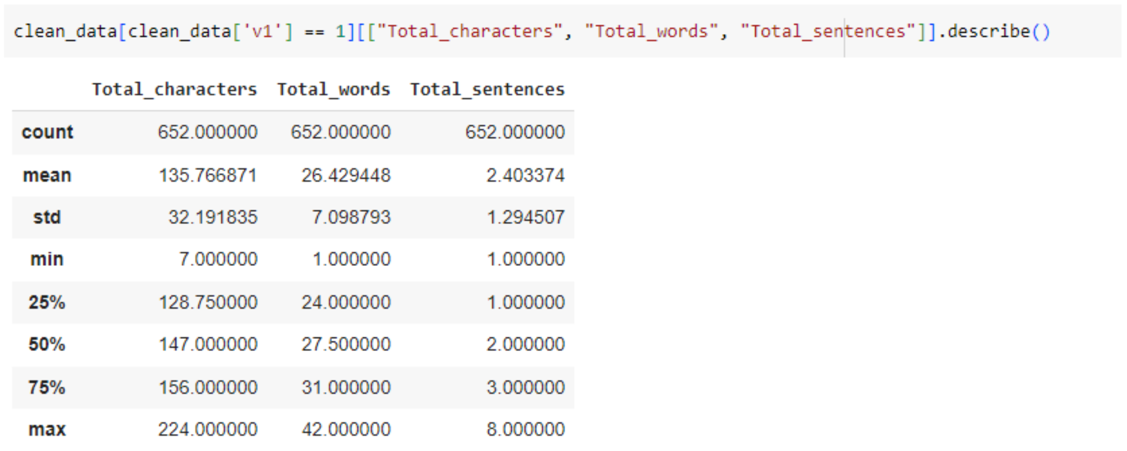


```
clean_data['Total_sentences'] = clean_data['v2'].apply(lambda x: len(nltk.sent_tokenize(x)))
clean_data
```



	v2	v1	Total_characters	Total_words	Total_sentences
0	Free entry in 2 a wkly comp to win FA Cup fina...	1	153	35	2
1	FreeMsg Hey there darling its been 3 weeks now...	1	145	34	2
2	WINNER As a valued network customer you have b...	1	157	29	3
3	Had your mobile 11 months or more? U R entitle...	1	153	30	2
4	SIX chances to win CASH From 100 to 20,000 pou...	1	134	30	2
...	...	...	...	...	...
931	2 new message	0	13	3	1
932	your bill is here	0	17	4	1
933	please check your	0	18	3	1
934	email has been reposted , regulation	0	36	6	1
935	request to deactivate	0	21	3	1

Table 12 Tokenization of the dataset using sent tokenizer.



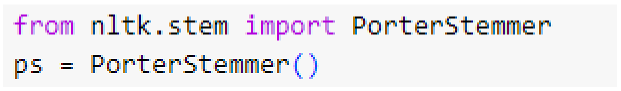
```
clean_data[clean_data['v1'] == 1][["Total_characters", "Total_words", "Total_sentences"]].describe()
```

	Total_characters	Total_words	Total_sentences
count	652.000000	652.000000	652.000000
mean	135.766871	26.429448	2.403374
std	32.191835	7.098793	1.294507
min	7.000000	1.000000	1.000000
25%	128.750000	24.000000	1.000000
50%	147.000000	27.500000	2.000000
75%	156.000000	31.000000	3.000000
max	224.000000	42.000000	8.000000

Table 13 Descriptive statistics of the tokenized sentences

4.3.5.2 Stemming

Stemming is a natural language technique, reduces and modifies words into their root forms, hence improving text preprocessing according to Nebojsa Bacanin (2023) To build a robust model it is necessary to normalize texts by removing repetitive words and to transform words into their base form through stemming. It can be applied to different forms of information retrieval, text mining, as well as email classification. There are different types of stemming such as porter stemmer, snowball stemmer, lancaster stemmer, and regex stemmer. Porter stemmer gives a resultant stem in a shorter word with the same root meaning, the porter stemmer was subsequently used in this research. The porter stemmer is a module in NLTK and it is imported using the code in figure 30 below.



```python
from nltk.stem import PorterStemmer
ps = PorterStemmer()
```

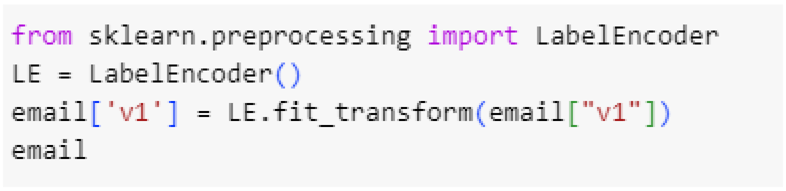
Figure 30 Stemming using Porter Stemmer

4.3.5.3 Stop Words

Stop words, in the context of Natural Language Processing (NLP), refer to English words that contribute minimally to the overall meaning of a sentence. Typically, these words are excluded from Natural Language data processing, as they often constitute the most common and less meaningful terms in a language from the perspective of machine learning models. Common examples of stop words include "the," "is," "at," "which," and "on". In contrast to tasks like language translation, stop words offer limited utility for our model, thus warranting their exclusion from the corpus. One notable advantage of removing stop words is the reduction in training time, with little impact on model accuracy. Moreover, this process enhances performance by retaining a more focused set of significant tokens, ultimately leading to improved classification accuracy. However, it's essential to exercise caution when removing stop words, as improper selection and removal can alter the text's intended meaning. Python offers several libraries, such as NLTK, SpaCy, and Gensim packages, which facilitate the removal of stop words while maintaining the text's integrity and improving the efficiency of NLP tasks.

4.3.6 Label encoding

Label encoding is a technique used to convert categorical variables into numerical variables suitable for the machine learning model. It converts all the columns in each table from a categorical column into a numerical column which can be fitted into the model. It is a vital preprocessing stage in a machine learning project. In this research, the email class was converted into a numerical column where each email category is denoted by 0 and 1. In this research the categorical value was replaced with a numerical value between 0 and 1, where 0 stands for ham, and 1 stands for spam for the spam/ham dataset. 0 stands for phishing, and 1 for ham for the ham/phishing dataset. And 0 for spam and 1 stands for phishing in the spam/phishing dataset. The figure 31 below illustrates the python code used for label encoding of the categorical variables (Nebojsa Bacanin, 2023)



```python
from sklearn.preprocessing import LabelEncoder
LE = LabelEncoder()
email['v1'] = LE.fit_transform(email["v1"])
email
```

Figure 31 Data Preprocessing using Label encoder.

4.3.7 Bags of Words.

The intrinsic disorderliness of textual data makes it unsuitable for direct use in machine learning techniques. As a result, there is a pressing need to transform text data into numerical representations, specifically vectors. Feature extraction or feature encoding are frequent names for this processing procedure. The "Bag of Words" (BoW) approach is one of the most used feature extraction methods for text data. BoW is a crucial modeling tool, especially when it comes to the use of machine learning techniques. It performs the role of a textual representation by accurately identifying word occurrences in a text and evaluating the vocabulary and usage of known terms.

4.3.8 Text vectorization TF-IDF

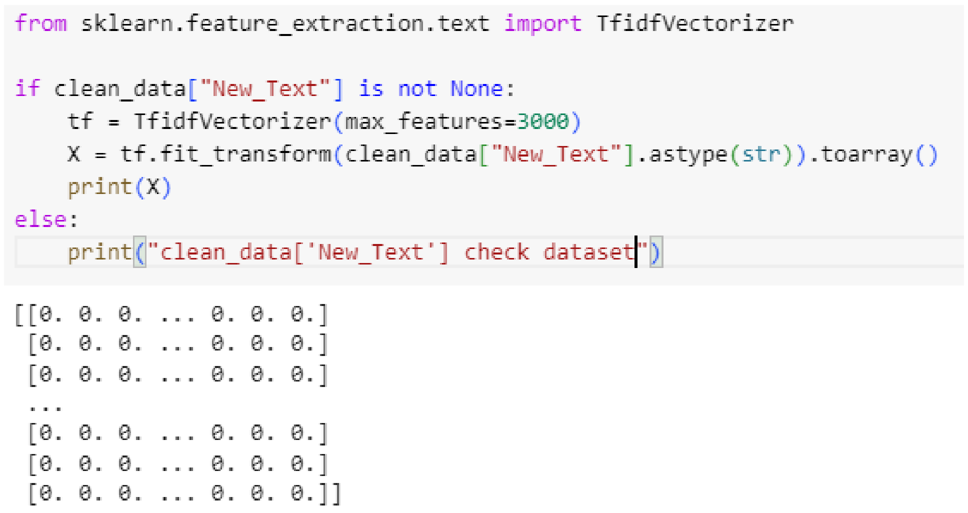
Highly frequent words tend to dominate the document hence leading to larger score, but they may not contain the needed information to the model, Term Frequency-Inverse Document Frequency is an approach used to rescale the frequency of words based on how often they appear in all documents. While Term Frequency refers to scoring of words in the current document, the weight of a term that occurs in a document is proportional to the term frequency and is illustrated below.
Tf(t,d) = count of t in d / number of words in d.
Document frequency tests the meaning of the text and it's very similar to TF, in the corpus. While term frequency is the frequency counter for a term t, df is the number of occurrences in the document set N. Df(t) - occurrence of t in documents. Inverse Frequency scores how rare and relevant the word is across the documents. The aim is to locate the appropriate records that fit the demand. It is computed using the TfidfVectorizer() method in Sklearn. The figure below shows an array of the frequency of the cleaned email dataset. Frequently occurring words often dominate a document, potentially inflating their importance, although they might not necessarily convey essential information to the model. To address this issue, Term Frequency-Inverse Document Frequency (TF-IDF) is a valuable approach for rescaling word frequencies based on their prevalence across all documents. Term Frequency (TF) quantifies word occurrences within a specific document, calculated as follows:

$$TF(t,d) = \frac{count\ of\ t\ in\ d}{number\ of\ words\ in\ d}$$

Equation 7 Text Vectorization using TF-IDF

Document Frequency (DF) assesses the significance of a term in the corpus by counting its occurrences in the document set DF(t)=occurrence of t in documents. Inverse Frequency (IDF) gauges a term's rarity and

relevance across documents, facilitating the identification of pertinent records. This computation is typically achieved using the TfidfVectorizer() method available in libraries like Scikit-Learn.
The relationship between TF and IDF enables the generation of TF-IDF scores, rebalancing word importance in text data. These concepts are illustrated in the figure 32.0 below, showcasing the frequency distribution within the cleaned email dataset.


```
from sklearn.feature_extraction.text import TfidfVectorizer

if clean_data["New_Text"] is not None:
    tf = TfidfVectorizer(max_features=3000)
    X = tf.fit_transform(clean_data["New_Text"].astype(str)).toarray()
    print(X)
else:
    print("clean_data['New_Text'] check dataset")
```

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

Figure 32 Text Vectorization using TFIDF Vectorizer.

4.3.9 Feature selection

In machine learning, feature selection is the act of selecting a subset of useful characteristics from a dataset while removing unimportant or redundant ones. This improvement increases computational effectiveness, lowers overfitting, and improves model performance. Filter methods (such as correlation analysis), wrapper methods (such as forward selection), and embedding methods (such as the feature importance of Random Forest) are the three basic techniques for feature selection. While wrapper techniques and embedded methods incorporate feature selection into the model training process, filter approaches assess feature relevance separately (Guyon & Elisseeff, 2003).

4.3.10 Feature Importance

feature importance measures how much each input feature affects model predictions. Each aspect is given a score, with higher values indicating more effect. By exposing feature-target linkages and enhancing models through dimensionality reduction, this method helps users better understand their data and make model conclusions more understandable. By keeping the most informative properties and enhancing their functionality, it aids in model optimization (Hastie et al., 2009).

4.3.11 Dimensionality reduction

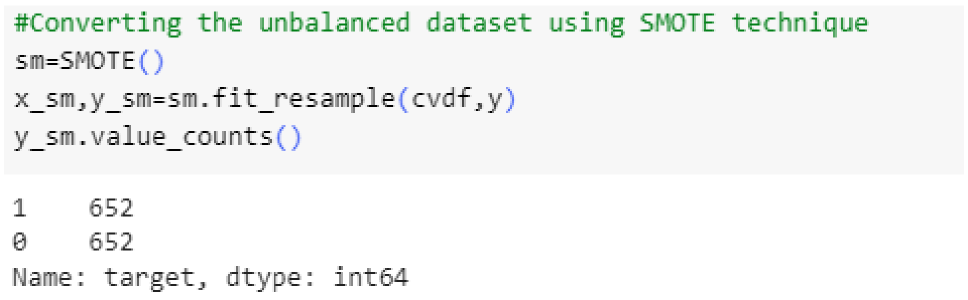
New features added to a machine learning problem adds to its complexity, hence the need for dimensionality reduction, a set of method used to remove irrelevant and excessive features from a machine learning model which reduces storage, computational time, model accuracy, improved training time, and removes noise from a dataset. Dimensionality reduction removes features that are not important in the data or affecting model accuracy. In the project irrelevant features like the date column, the subject of the email, sender, and receiver were removed before training the model.

4.3.12 Feature Extraction

Feature extraction transforms textual data into machine learning-friendly features for improved performance. Since machine learning algorithms do not accept raw text data directly, sentences and words must first be converted into numbers or vectors called features. The Bag-of-Words model is utilized as a feature extractor in Natural Language Processing and Information Retrieval. It considers word frequency in emails as a feature, rather than word order or position in phrases. Bag-of-Words was used in this research to identify the word frequency.

4.3.13 Imbalanced data (SMOTE)

Machine learning models frequently struggle with datasets exhibiting notable class imbalances. Synthetic Minority Oversampling Technique, or SMOTE, is one efficient method for resolving this problem. By creating synthetic samples for the minority class inside the training dataset, SMOTE tackles the issue of class imbalance. SMOTE achieves this by duplicating existing examples from the minority class, allowing the creation of as many synthetic instances as needed (Nebojsa Bacanin, 2023). These synthetic instances are strategically crafted to closely resemble the characteristics of the original minority class examples. The implementation of SMOTE was facilitated through the Python library "imbalanced-learn" integrated into the development environment. This technique is instrumental in rebalancing class distributions, enhancing model performance, and ensuring that the classification model effectively learns from both majority and minority class instances.



```
#Converting the unbalanced dataset using SMOTE technique
sm=SMOTE()
x_sm,y_sm=sm.fit_resample(cvdf,y)
y_sm.value_counts()
```

```
1    652
0    652
Name: target, dtype: int64
```

Figure 33 Data Preprocessing using SMOTE.

4.4 Machine learning algorithms.

The aim of this project is to create a robust email classifier capable of accurately categorizing incoming emails as either spam, ham (legitimate), or phishing. To assess the effectiveness of various machine learning algorithms for this email classification task, computational experiments were conducted. Three distinct algorithms were employed namely Multinomial Naive Bayes, Support Vector Machine, XGBoost. Performance evaluation was carried out using a confusion matrix, enabling the calculation of essential metrics such as accuracy, precision, recall, and F-measure. These metrics collectively gauge the efficiency and reliability of the developed email classification model.

4.4.1 Support Vector Machine (SVM)

SVM represents a supervised machine learning algorithm adept at tackling both classification and regression tasks. Predominantly, SVM finds its application in classification challenges. This algorithm takes raw data as input and, through a mathematical process, yields a discriminative line that effectively separates distinct classes, as elaborated by Nebojsa Bacanin (2023). One of SVM's distinguishing characteristics is its ability to map data into a higher-dimensional feature space, a technique that enables the categorization of data points, even when linear separability is not apparent. The separating boundaries in this space are known as hyperplanes. SVM's objective is to locate the hyperplane that maximizes the margin, signifying the greatest distance between the two classes. SVM encompasses two primary variants: linear SVM, designed for linearly separable data, and non-linear SVM, tailored to handle complex, non-linearly separable datasets. In the context of this research, SVM exhibited superior performance metrics. It achieved this by generating hyperplanes to differentiate various classes based on the distinct features derived from the dataset.

4.4.2 Naïve Bayes

The Naïve Bayes classifier is a supervised machine learning technique primarily employed for text classification tasks. It derives its foundation from probabilistic classifiers, utilizing the Bayesian theorem as its core principle. This method operates under a fundamental assumption: the presence of any particular feature within a class is conditionally independent of the presence of any other feature, given the class. In other words, it assumes that features contribute to classification independently. For this specific project, two variants of the Naïve Bayes algorithm were used: Multinomial Naïve Bayes: This probabilistic learning method, commonly used in Natural Language Processing (NLP) tasks, calculates the probability of each tag for a given sample. Subsequently, it assigns the tag with the highest calculated probability as the output category. Gaussian Naïve Bayes: Gaussian Naïve Bayes is another variant of the Naïve Bayes algorithm. It is typically applied to datasets with continuous numerical features and assumes that these features follow a Gaussian (normal) distribution. Like the Multinomial variant, Gaussian Naïve Bayes calculates probabilities based on the Gaussian distribution to make classification decisions. In essence, the Naïve Bayes classifier offers an efficient and probabilistic means of categorizing text data, making it particularly well-suited for NLP tasks and text classification challenges as noted by (Kotsiantis et al., 2006).

$$p\left(\frac{B}{A}\right) = \left(\frac{p(A \cap B)}{P(B)}\right)$$

Equation 8 Naive Bayes Equation

Where
PA stands for the prior probability of occurring A
PBA stands for the condition probability of B given that A occurs.
PAB stands for the condition probability of A given that B occurs.
PB stands for the probability of B occurring.

4.4.3 Xgboost

XGBoost, also known as Extreme Gradient Boosting, is a robust and scalable machine learning algorithm that belongs to the family of gradient boosting frameworks. This algorithm excels in delivering efficient and highly accurate predictions across a wide spectrum of tasks, including regression, classification, and ranking problems. The core idea behind XGBoost revolves around boosting—a technique that strengthens a single, relatively weak model by amalgamating it with multiple other weak models (Chen & Guestrin, 2016). The goal is to collectively create a robust and high-performing model.

This process unfolds by focusing on the expected outcomes for the next model iteration, with the aim of minimizing prediction errors. These expected outcomes are determined based on the gradient of the error concerning the current prediction. XGBoost operates iteratively, training shallow decision trees in each iteration. Crucially, each iteration leverages the error residuals of the previous model to fine-tune the next model. The final prediction is an intelligently weighted sum of predictions from all the individual trees. This approach not only enhances predictive accuracy but also serves as a mechanism to mitigate bias and underfitting, thereby ensuring the model's robustness and generalizability.

4.4.4 Random Forest

Random Forest is a widely used machine learning algorithm used in supervised learning. According to Paul et al. (2018), this system is renowned for its competency and excels at resolving data imbalances across several classes, particularly in situations involving large datasets. The fundamental idea of ensemble learning, a technique that combines the predictive power of various classifiers to produce a more precise and reliable model, is at the core of Random Forest. It does this by using a group of decision trees, each of which was built using a different subset of the dataset. To improve the model's overall predicted accuracy, it then integrates the individual predictions from these trees, frequently taking their average. In the context of this study, a notable example entails the construction of an ensemble made up of 100 decision trees. This careful selection of a sizable number of trees supports the overriding objective of improving the email categorization model's performance and accuracy. Email classification and other machine learning applications benefit from Random Forest's ability to avoid overfitting issues while also fostering prediction precision by utilizing the collective judgment of several decision trees.

4.5 Test/ training of the machine learning model

4.5.1 Train Model

Training a machine learning model for spam-ham-phishing classification is a crucial step in developing an effective email filtering system. It involves dividing the dataset into two parts: the Training Set and the Testing Set. The Training Set is where the machine learning algorithm learns to distinguish between spam, ham (legitimate emails), and phishing emails. During this process, the algorithm analyzes various features, such as email content, sender information, and subject lines, to identify patterns and relationships that differentiate these email types. The quality of the Training Set is vital, as it directly affects the model's accuracy. A well-structured and diverse Training Set enhances the model's ability to classify emails correctly. In this research, a common practice is to allocate about 70% of the dataset to the Training Set, while the remaining 30% becomes the Testing Set. These subsets are organized as "X-train," "X-test," "Y-train," and "Y-test," facilitating data segregation for training and evaluation. Once the model is trained, it gains the knowledge and insights needed for accurate predictions. It can then classify incoming emails into spam, ham, or phishing categories based on the patterns and information it has learned. The Training Set represents the foundation upon which the machine learning model's predictive abilities are built, ultimately contributing to more effective email classification and improved email security.

4.5.2 Test model

After the successful training of the machine learning model on the dataset, the next critical step is the testing phase. This stage serves as a litmus test for the model's proficiency and its ability to generalize effectively when presented with new or unseen data. In the context of this research, approximately 30% of the original dataset was reserved for testing purposes. This partitioning is crucial to ensure that the model's performance is rigorously assessed on data it has never encountered during training. The testing dataset is conventionally designated as "X_test" (representing input features) and "Y_test" (representing target labels). During the testing phase, the model is exposed to the "X_test" data, allowing it to make predictions based on its learned patterns and relationships. These predictions are then compared to the ground truth labels provided by "Y_test." This comparison serves as the basis for evaluating the model's performance. Various performance metrics, including accuracy, precision, recall, and F1 score, are often used to gauge the model's effectiveness in correctly classifying data in real-world scenarios. These metrics provide insights into the model's ability to make accurate predictions and its capacity to generalize beyond the training data.

4.6 Evaluation

In machine learning, model evaluation is a crucial step to determine how well a trained model performs in real-world scenarios. To gauge the effectiveness of a proposed method, a comprehensive set of eight metrics is employed. These metrics encompass both the parameters of the confusion matrix, which include True Positives, False Positives, True Negatives, and False Negatives, as well as key classification outcomes. The F1 score, a balanced measure of precision and recall, provides insights into overall model performance. Accuracy quantifies the model's correctness, while precision assesses its ability to avoid false positives. Recall measures the model's sensitivity to detecting positive cases. Users can evaluate the model's capacity to differentiate between spam, ham, and phishing emails using this comprehensive evaluation approach. They are better able to understand the model's strengths and potential improvement areas by taking into account a variety of measures, which enables analysts to make adept choices about how to improve its performance.

Predicted class		Actual Class	
		Yes	No
	Yes	TP	FN
	No	FP	TN

Table 14 Confusion Matrix showing the Positives and Negatives

4.6.1 True Positive rate (TP)

It is a performance metric used to denote the percentage of spam messages that were classified by the machine learning model.it is illustrated as the total number of spam messages divided by the number of spam messages accurately classified.

$$TP = \frac{P}{S}$$

Equation 9 True Positives Rate

where S is the total number of spam messages, and P is the predicted spam messages.

4.6.2 True Negative Rate (TN)

It is defined as the total number of non-spams divided by the number of non-spams predicted by the model. True Negatives denote the percentage of non-spam messages accurately predicted as non-spam by the machine learning model.

$$TN = \frac{Q}{N}$$

Equation 10 True Negative Rate

4.6.3 False Positive Rate (FP)

False Positives occur in a model during classification where the machine learning algorithm misclassifies or wrongly categorizes non-spam messages as spam messages. E.G if non-spam messages are denoted N, and the misclassified non-spam messages as M. The illustration is highlighted below.

$$FN = \frac{M}{N}$$

Equation 11 False Positive Rate

4.6.4 False Negatives rate (FN)

It indicates the proportion of spam messages that the machine learning algorithm misclassified as non-spam messages. False negatives misclassify the spam messages thereby wrongly classifying them as non-spam messages. The formula used to indicate the percentage of FN is illustrated in equation 12 below.

$$FN = \frac{T}{S}$$

Equation 12 False Negatives Rate

4.6.5 Precision

It indicates the proportion of messages that the machine learning algorithm categorized as spam. It demonstrates absolute correctness of the model. It is denoted by the formula below.

$$precision = \frac{TP}{TP+FP}$$

Equation 13 Precision Rate

4.6.6 Recall

This shows the measure of completeness of the model. It denotes the percentage of messages that were spam and classified as spam.

$$Recall = \frac{TP}{TP+FN}$$

Equation 14 Recall Rate

4.6.7 F1-score

It measures the harmonic means of precision and recall. The formula is illustrated in the equation below.

$$F1\ Score = 2 * \frac{Precision*Recall}{Precision + Recall}$$

Equation 15 F1-Score Rate

4.6.8 Accuracy

Accuracy is a performance index used to measure classification algorithms. It is calculated as the ratio of correctly predicted samples to the total number of test samples. Accuracy is a good measure to determine how well the model classified the data. For an imbalanced dataset, they tend to Favor the dataset with the highest number of non-spam emails. An example is predicting a classification model where non-spam is 98 and spam is 2, the accuracy of the prediction would be 98% by predicting all samples as non-spam, while failing to effectively recognize spam.

4.7 Result

In the result below, text classification was initially performed using the Term Frequency Inverse Document Frequency approach. Predicting the outcome of the email classification technique, 5 different machine learning algorithms were implemented to perform a comparative analysis of its performance using different key performance metrics such as accuracy, precision, recall, and F1 score. Confusion matrix was also developed to highlight regions in which the model failed to correctly classify data during testing. The following is a summary of the machine learning algorithm utilized in this study: Support Vector Machine.

Gaussian Naive Bayes, Multinomial Naive Bayes, Random Forest and XGBoost. The algorithm was implemented on the 3 different datasets namely spam-ham.csv, Spam-phishing.csv, ham-phishing.csv. The result is divided into three sections. The first section shows the performance index for the non-smote for each of the machine learning algorithms used.

4.7.1 Section 1

This section gives an exploration of the different email types, presenting the outcomes of machine learning algorithms utilizing the Non-SMOTE technique for three distinct email categories (Spam-Ham. Ham-Phishing, Spam-Phishing). The evaluation encompasses five performance indices, namely SVC, XGBoost, Multinomial, Random Forest, and Gaussian Naïve Bayes. This structured approach facilitates a comprehensive comparison of the machine learning models' effectiveness in classifying diverse email types, offering valuable insights into their performance across multiple dimensions.

4.7.1.1 Spam-Ham (Non-SMOTE) For SVC

In the result below, the performance index for the non-smote technique using a support vector machine shows the classification of spam to be 0 and ham denoted by 1. In the imbalance dataset, the model prediction tends to favor the spam class with precision, recall, F1 score showing value of 91%, 99%, 95% respectively and an accuracy of 91%. For ham prediction, the model generally had a poor performance due to the majority class prediction of spam. However, the macro avg gives the average prediction of the model considering both factors as illustrated in table 15.0 below.

	Precision	Recall	F1-Score	Support

0	0.91	0.99	0.95	1345
1	0.85	0.35	0.50	206
Macro avg	0.88	0.67	0.72	1551

Table 15 Classification Report for Spam-Ham (Non-Smote) SVC Algorithm

The confusion matrix also shows that the majority of the predictions favor the TP (True positives) due to the imbalance class categories. In figure 34.0 below, the predicted and actual label for SVM model classified has spam giving a greater classification of 1,332, which is over 85% of the entire classification.

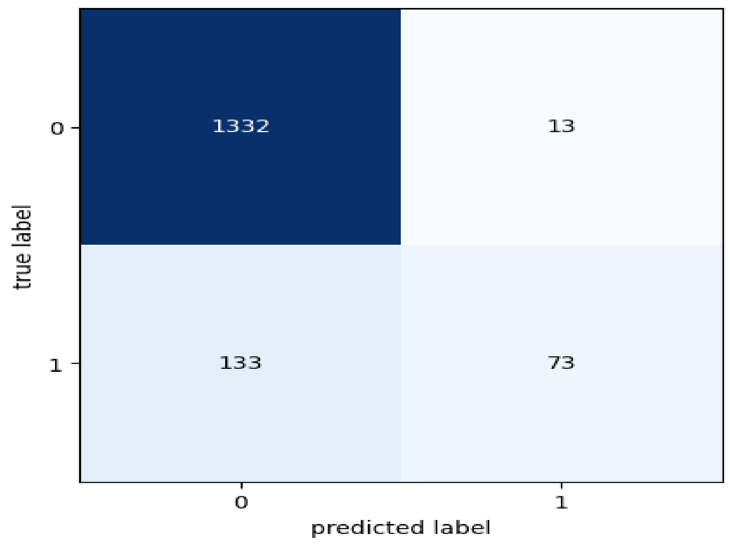


Figure 34 Confusion Matrix for Spam-Ham (Non-Smote) SVC

4.7.1.2 Spam-Ham (Non-SMOTE) For XGBoost

The XGBoost classifier for spam-ham dataset using a non-smote technique, shows the macro avg performance index having precision of 87%, recall 61%, F1-score of 64% and accuracy of 89%, however its prediction performed also favored the majority class consisting of spam set. its prediction for ham gave a lesser value as compared to SVC.

	Precision	Recall	F1-Score	Support
0	0.89	0.99	0.94	1345
1	0.85	0.22	0.35	206

Macro avg	0.87	0.61	0.64	1551

Table 16 Classification Report for Spam-Ham (Non-Smote) XGBoost

The confusion matrix below shows the True positive value having 1,337 classes, False positives with 8 classes, false negative with value 161 and true positive with 45 categories. The model has a lesser false positive, but a greater false negative i.e predicting the ham as spam which can lead to false classification in production.

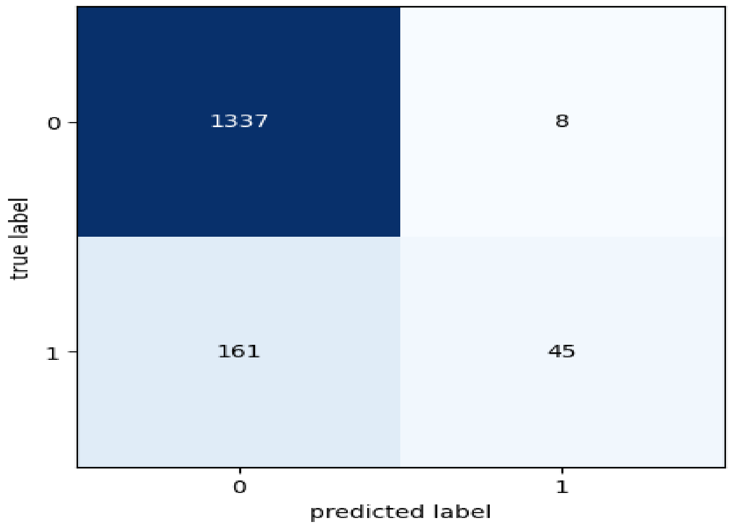


Figure 35 Confusion Matrix for Spam-Ham (Non-Smote) XGBoost

4.7.1.3 Spam-Ham (Non-SMOTE) For Multinomial NB

Using the Multinomial Naive Bayes classification algorithm, the model performed better in the spam class and under performed in the ham class category. In figure 9.0 below, the defaulter's class has a precision of 97%, recall of 19%, F1-score of 32% and an overall accuracy of 89%. The macro avg shows the overall model performance to be 93% precision, 59% recall and 63% F1-score.

	Precision	Recall	F1-Score	Support
0	0.89	1.00	0.94	1345
1	0.97	0.19	0.32	206
Macro avg	0.93	0.59	0.63	1551

Table 17 Classification Report for Spam-Ham (Non-Smote) MNB.

The confusion matrix indicates a higher false negative of 167, true positive rate also increases to 1344 a lesser false positive value of 1 and a true negative of 39 for the non-smote category as shown in figure 18.0 below.

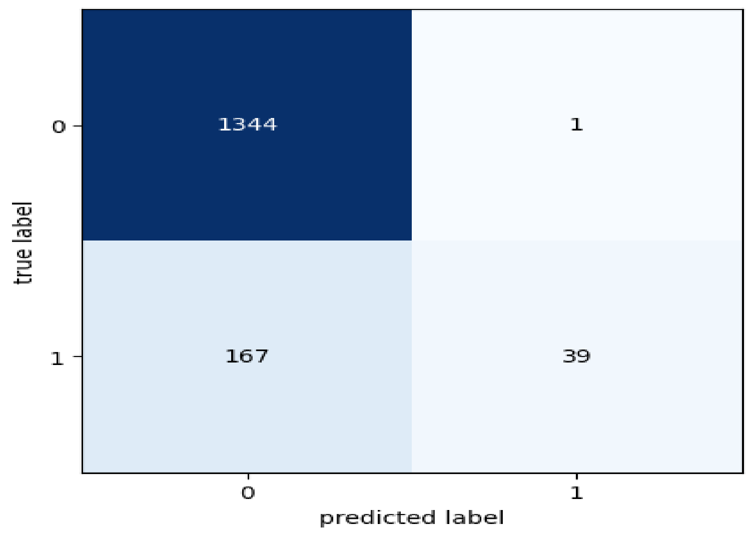


Figure 36 Confusion Matrix for Spam-Ham (Non-Smote) MNB

Fig 18.0 shows the confusion matrix of Spam-Ham (Non-SMOTE) For Multinomial NB

4.7.1.4 Spam-Ham (Non-SMOTE) For Random Forest Classifier

The classification report of the random forest classifier is illustrated in table 18.0. The table below shows the result of the classification for spam and ham class using non-smote. The model accuracy was 91% and the macro avg scores for precision, recall, F1-score was 88%, 67% and 72% respectively.

	Precision	Recall	F1-Score	Support
0	0.91	0.99	0.95	1345
1	0.85	0.35	0.50	206
Macro avg	0.88	0.67	0.72	1551

Table 18 Classification Report for Spam-Ham (Non-Smote) Random Forest Classifier

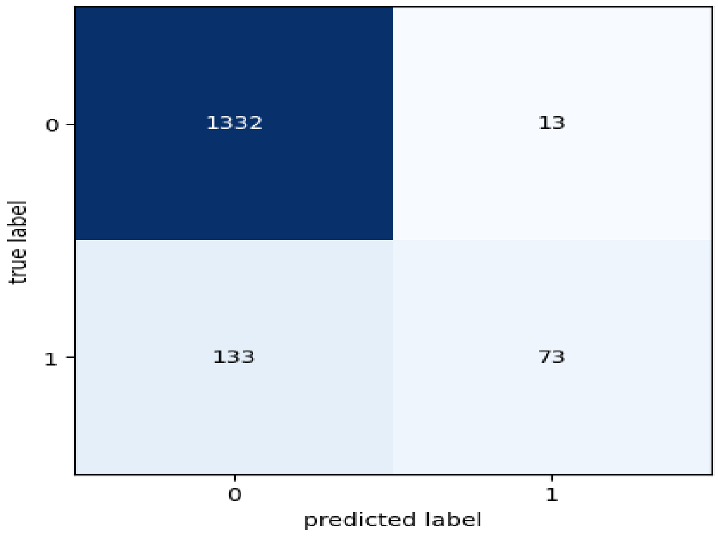


Figure 37 Confusion Matrix for Spam-Ham (Non-Smote) Random Forest Classifier

4.7.1.5 Spam-Ham (Non-SMOTE) For Gaussian NB

The result for the Gaussian NB classification shown below denoted the macro avg for precision, recall and F1-score to be 57%, 63% 40% respectively. In the confusion matrix in figure 38 below the false positive has a larger value of 887 denoting an increased misclassification, while TP value gave 458 and TN value gave 188 respectively.

	Precision	Recall	F1-Score	Support
0	0.96	0.34	0.50	1345
1	0.17	0.91	0.29	206
Macro avg	0.57	0.63	0.40	1551

Table 19 Classification Report for Spam-Ham (Non-SMOTE) GNB.

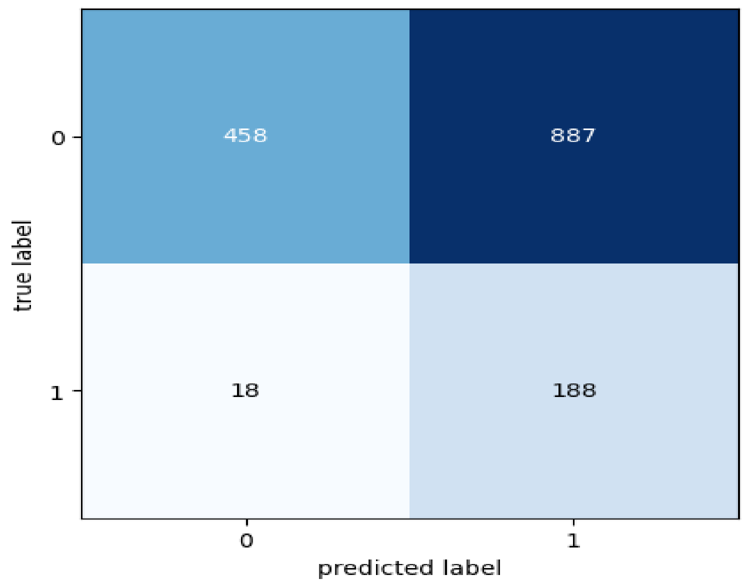


Figure 38 Confusion Matrix for Spam-Ham (Non-Smote) GNB

4.7.1.6 Spam-Phishing (Non-SMOTE) For SVC

In spam phishing classification report below, the macro avg has a precision value of 79%, recall 53%, F1-score 51% for the spam-phishing (non-smote) technique. The confusion matrix below shows the true and predicted label during the training of the machine learning model. The TP shows a value of 3 and the majority of the value was classified as TN. Although the model appears to poorly classify the phishing data, the accuracy of the model still gave 83%.

	Precision	Recall	F1-Score	Support
0	0.75	0.06	0.12	47
1	0.82	1.00	0.90	202
Macro avg	0.79	0.53	0.51	249

Table 20 Classification Report for Spam-Phishing (Non-SMOTE) SVC

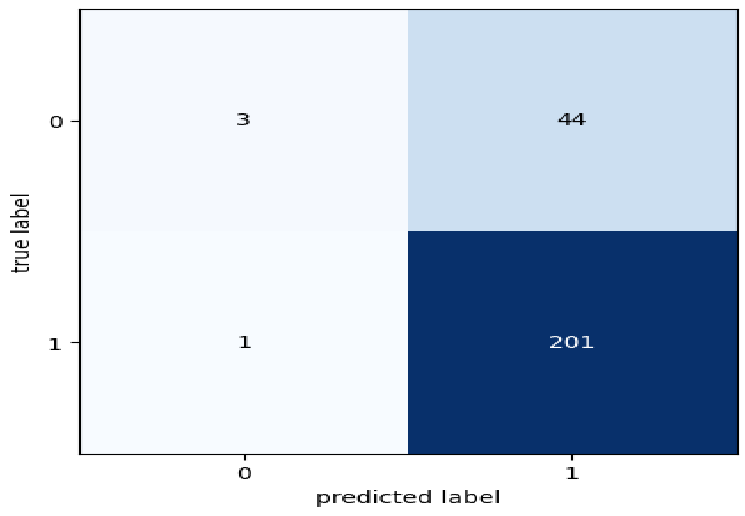


Figure 39 Confusion Matrix for Spam-Phishing (Non-Smote) SVC

4.7.1.7 Spam-Phishing (Non-SMOTE) For XGBoost

In the classification report below, the macro avg shows a lower performance index compared to the SVC for Non-smote technique. From the table 21.0 below, the macro avg for precision, recall and F1-score gave 66%, 51% and 47% respectively with an accuracy of 81%. The classification report further explains the model result if the spam and phishing class below. The confusion matrix in figure 40.0 shows the model performance using TP, TN, FP and FN. Majority of the class was classified as False Negative and a graphical illustrated is shown below.

	Precision	Recall	F1-Score	Support
0	0.50	0.02	0.04	47
1	0.81	1.00	0.90	202
Macro avg	0.66	0.51	0.47	249

Table 21 Classification Report for Spam-Phishing (Non-SMOTE) XGBoost

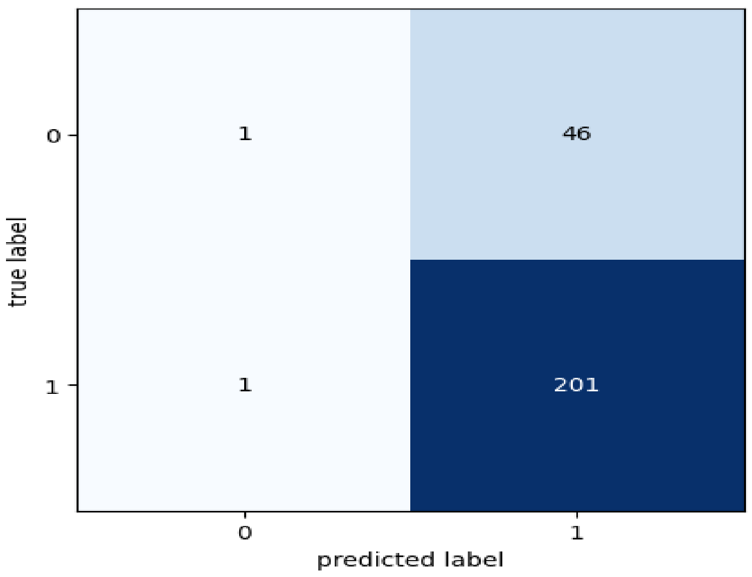


Figure 40 Confusion Matrix for Spam-Phishing (Non-Smote) XGBoost

4.7.1.8 Spam-Phishing (Non-SMOTE) For Multinomial NB

The classification report for Spam-phishing (non-smote) for Multinomial NB is shown below. Table 22.0 compares the individual class and the macro avg. The confusion matrix shows the classification according to the number of positives and negatives and is illustrated in figure 41.0.

	Precision	Recall	F1-Score	Support
0	0.75	0.06	0.12	47
1	0.82	1.00	0.90	202
Macro avg	0.79	0.53	0.51	249

Table 22 Classification Report for Spam-Phishing (Non-SMOTE) MNB

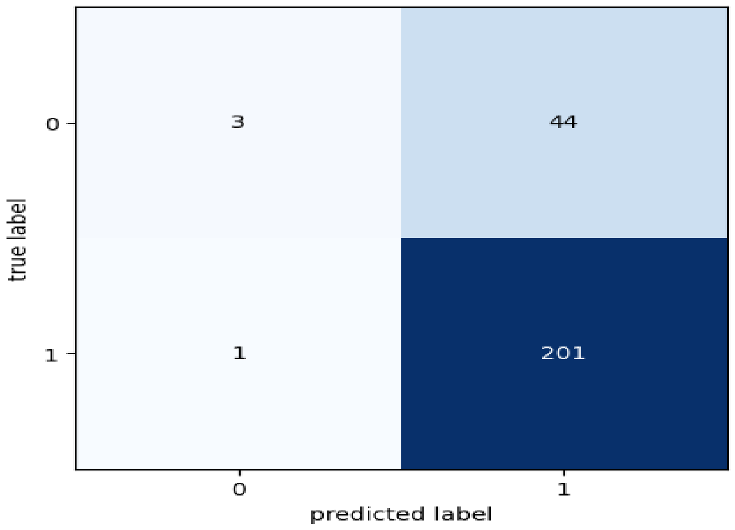


Figure 41 Confusion Matrix for Spam-Phishing (Non-Smote) MNB

4.7.1.9 Spam-Phishing (Non-SMOTE) For Random Forest Classifier

The Random Forest classification for non-smote spam-phishing dataset shows the macro avg to have an improved classification result as shown in table 23.0 below. The confusion matrix further illustrates how the model classified each set using the TP, TN, FP, and FN rates as shown in figure 42.

	Precision	Recall	F1-Score	Support
0	0.91	0.21	0.34	47
1	0.84	1.00	0.91	202
Macro avg	0.88	0.60	0.63	249

Table 23 Classification Report for Spam-Phishing (Non-SMOTE) Random Forest

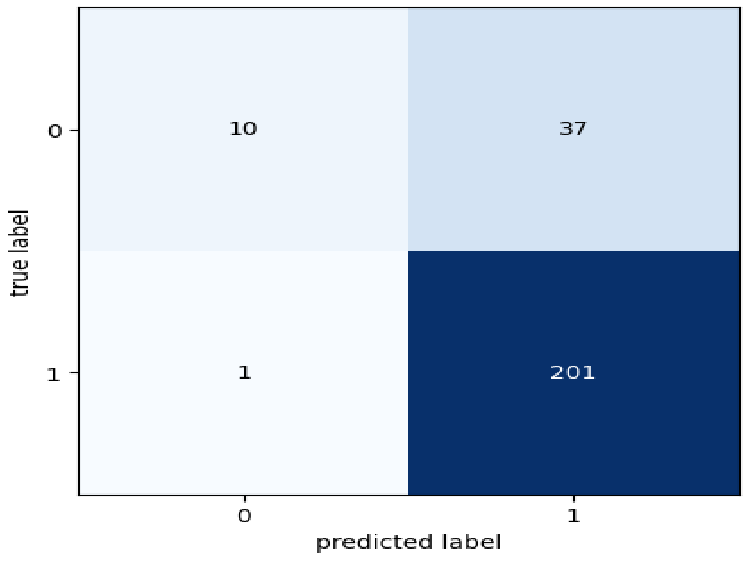


Figure 42 Confusion Matrix for Spam-Phishing (Non-Smote) Random Forest

4.7.1.10 Spam-Phishing (Non-SMOTE) For Gaussian NB

For Gaussian NB, which is good at handling continuous values shows the performance index on a spam-phishing dataset. The result shows the individual class performance during training as well as the macro avg with values of 63%, 70% and 53% for the precision, recall, and f1score respectively. The confusion matrix gave a fairly uniformed distribution, although there were a lot of false negatives, but the TP and TN gave a value of 45 and 91 respectively. The accuracy of the prediction was 55%.

	Precision	Recall	F1-Score	Support
0	0.29	0.96	0.44	47
1	0.98	0.45	0.62	202
Macro avg	0.63	0.70	0.53	249

Table 24 Classification Report for Spam-Phishing (Non-SMOTE) GNB

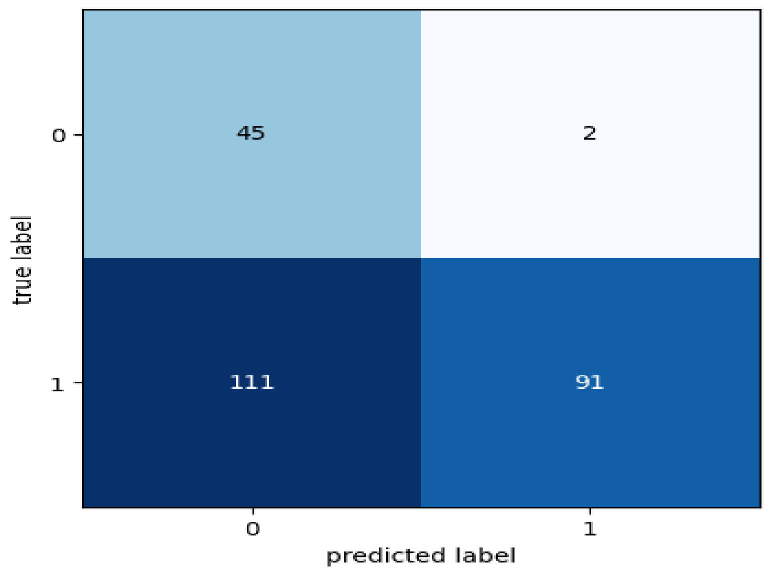


Figure 43 Confusion Matrix for Spam-Phishing (Non-Smote) GNB

4.7.1.11 Ham-Phishing (Non-SMOTE) For SVC

The dataset used for ham-phishing classification has the majority class as ham, meanwhile the classification report shows how the accuracy of the prediction is using other performance indexes. The accuracy of model prediction using SVC denoted 96%, with a macro avg value for precision, recall, f1-score of 86%, 55% and 58% respectively. The confusion matrix in figure 44 below majority of the class as TN with value 1349.

	Precision	Recall	F1-Score	Support
0	0.75	0.11	0.18	57
1	0.96	1.00	0.98	1351
Macro avg	0.86	0.55	0.58	1408

Table 25 Classification Report for Ham-Phishing (Non-SMOTE) SVC

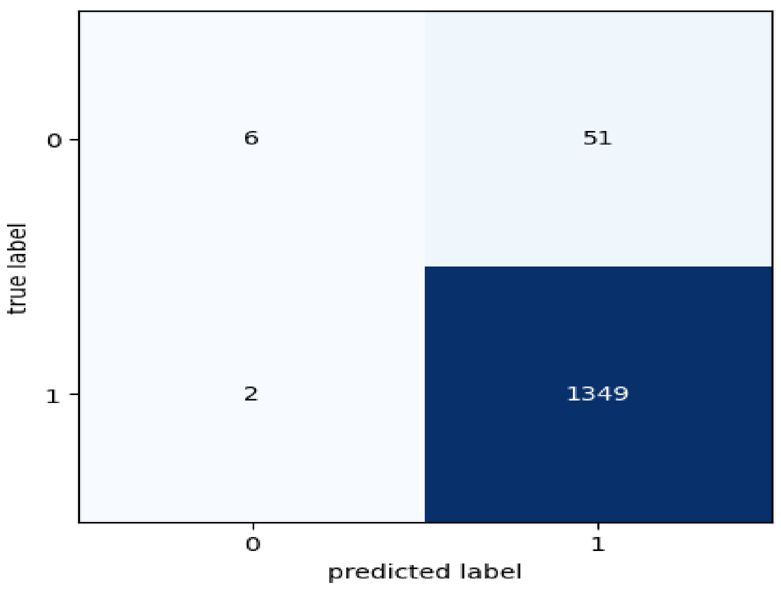


Figure 44 Confusion matrix for Ham-Phishing (Non-SMOTE) SVC

4.7.1.12 Ham-Phishing (Non-SMOTE) For Gaussian NB

Gaussian NB for Ham-phishing non-smote technique showed a reduced macro avg of 53% for precision, 68% for recall and 35% for recall. From table 26 below the percentage of correctly predicting the phishing class is relatively low as compared to other ML model. The accuracy of prediction was also very low with 42% accuracy. The confusion matrix shows the model wrongly predicted 809 class as FN and 2 as FP. The TP gave a value of 55 and TN value as 542.

	Precision	Recall	F1-Score	Support
0	0.06	0.96	0.12	57
1	1.00	0.40	0.57	1351
Macro avg	0.53	0.68	0.35	1408

Table 26 Classification Report for Ham-Phishing (Non-SMOTE) GNB

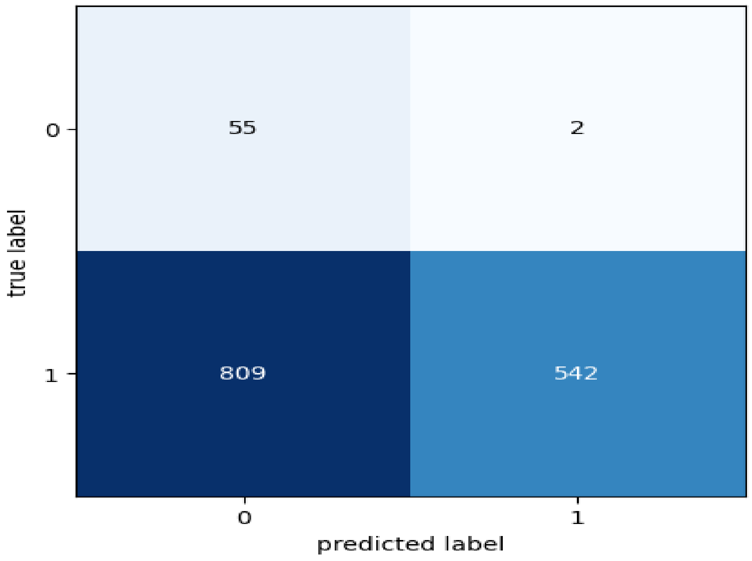


Figure 45 Confusion matrix for Ham-Phishing (Non-SMOTE) GNB

4.7.1.13 Ham-Phishing (Non-SMOTE) For Random Forest Classifier

In the table below, the classification report shows the performance index using a random forest classifier. Table 27 shows the percentage of precision, recall f1-score using the ham-phishing data, the result shows the macro avg for each performance index to be 90%, 63% and 69% respectively with an accuracy of 97%. The confusion matrix result is illustrated in figure 46.0 below.

	Precision	Recall	F1-Score	Support
0	0.83	0.26	0.40	57
1	0.97	1.00	0.98	1351
Macro avg	0.90	0.63	0.69	1408

Table 27 Classification Report for Ham-Phishing (Non-SMOTE) Random Forest

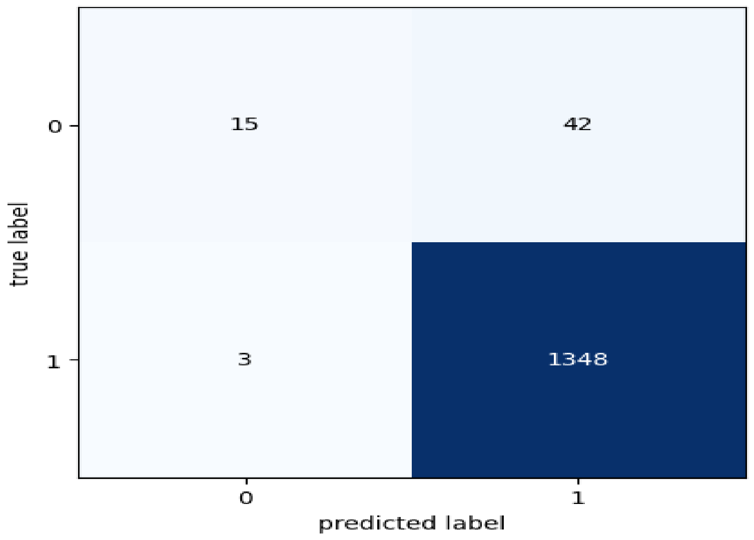


Figure 46 Confusion matrix for Ham-Phishing (Non-SMOTE) Random Forest

4.7.1.14 Ham-Phishing (Non-SMOTE) For XGBoost

In the research experiment below, the XGBoost classifier shows a macro avg of 48% for precision, 50% recall and 49% f1score. The reason for this low performance shows the phishing classes was not giving a null value for the index highlighted below. The confusion matrix also shows a significant increase in the number of TN with 1350. Accuracy of the model was 96%

	Precision	Recall	F1-Score	Support
0	0.00	0.00	0.00	57
1	0.96	1.00	0.98	1351
Macro avg	0.48	0.50	0.49	1408

Table 28 Classification Report for Ham-Phishing (Non-SMOTE) XGBoost.

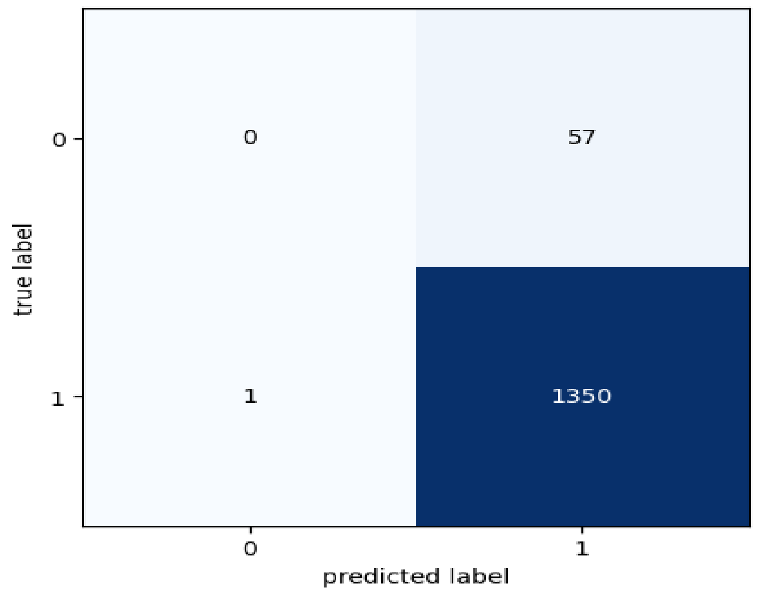


Figure 47 Confusion matrix for Ham-Phishing (Non-SMOTE) For XGBoost.

4.7.2 Section 2

This section employs the SMOTE technique to effectively classify various types of emails, such as spam-ham, spam-phishing, and ham-phishing, utilizing five distinct machine learning algorithms listed below.

4.7.2.1 Spam-Ham (SMOTE) For SVC

In the figure below the result shows the classification report for SVC for both the spam-ham emails. Using the macro average the precision, recall and F1_score all had a value of 96%. The confusion matrix is also illustrated in figure 48 below.

	Precision	Recall	F1-Score	Support
0	0.98	0.94	0.96	899
1	0.94	0.98	0.96	908
Macro avg	0.96	0.96	0.96	1807

Table 29 Classification Report for Spam-Ham (SMOTE) SVC

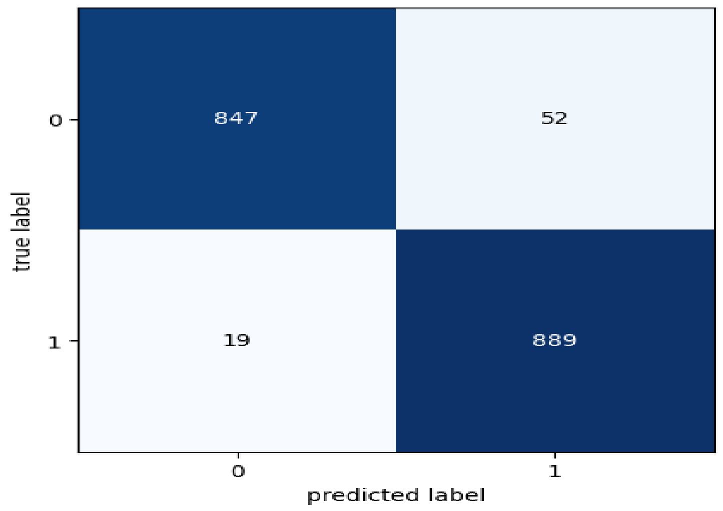


Figure 48 Confusion matrix for Spam-Ham (SMOTE) using SVC.

4.7.2.2 Spam-Ham (SMOTE) For XGBoost

XGBoost was used for classifying spam and ham dataset using the smote technique, the result is illustrated in the table below. The confusion matrix also shows how well the model performed and is described in figure 49 below.

	Precision	Recall	F1-Score	Support
0	0.98	1.00	0.99	899
1	1.00	0.98	0.99	908
Macro avg	0.99	0.99	0.99	1807

Table 30 Classification Report for Spam-Ham (SMOTE) XGBoost.

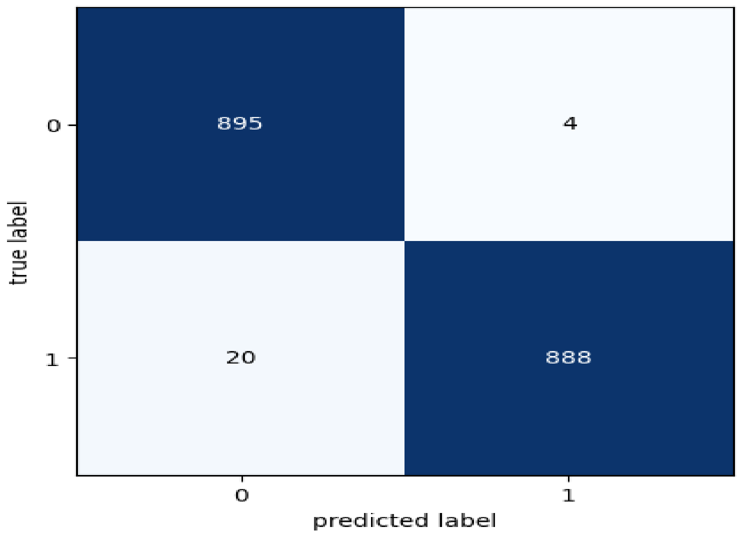


Figure 49 Confusion matrix for Spam-Ham (SMOTE) using XGBoost

4.7.2.3 Spam-Ham (SMOTE) For Multinomial NB

Using the Multinomial Naïve Bayes approach, the table 31 illustrates the different classification report for both the Spam and ham class using the smote technique. The confusion matrix also illustrates the positives and negative classifications.

	Precision	Recall	F1-Score	Support
0	0.82	0.98	0.89	899
1	0.98	0.78	0.87	908
Macro avg	0.90	0.88	0.88	1807

Table 31 Classification Report for Spam-Ham (SMOTE) MNB

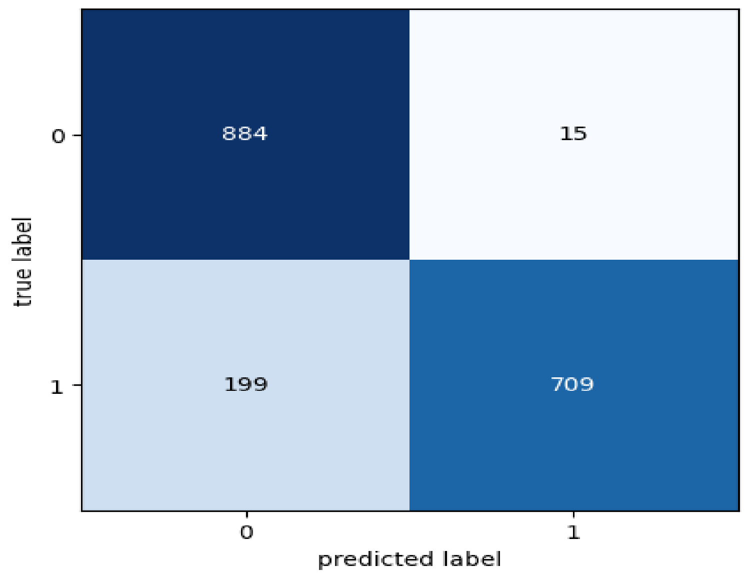


Figure 50 Confusion matrix for Spam-Ham (SMOTE) using MNB.

4.7.2.4 Spam-Ham (SMOTE) For Random Forest Classifier

Random Forest Classifier was also considered for this type of email class, and the result shows that the macro average has a precision, recall and f1-score of 96%. The confusion matrix in figure 51 illustrates an improved classification for the True Positives and True Negatives.

	Precision	Recall	F1-Score	Support
0	0.97	0.95	0.96	899
1	0.95	0.97	0.96	908
Macro avg	0.96	0.96	0.96	1807

Table 32 Classification Report for Spam-Ham (SMOTE) Random Forest.

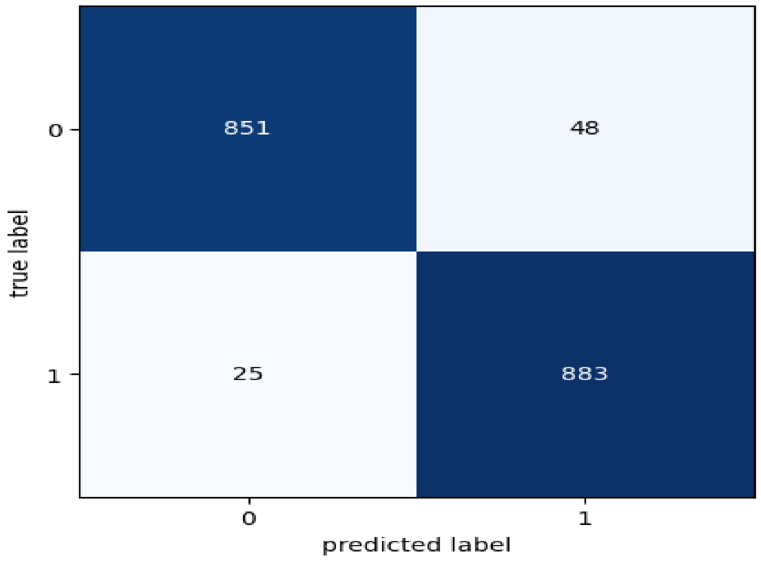


Figure 51 Confusion matrix for Spam-Ham (SMOTE) using Random Forest Classifier

4.7.2.5 Spam-Ham (SMOTE) For Gaussian NB

GNB has a macro average classification of 95% from table 33. The classification report shows the model has 823 True Positives and 894 true negatives classifying more customers has defaulters compared with other category.

	Precision	Recall	F1-Score	Support
0	0.98	0.92	0.95	899
1	0.92	0.98	0.95	908
Macro avg	0.95	0.95	0.95	1807

Table 33 Classification Report for Spam-Ham (SMOTE) GNB

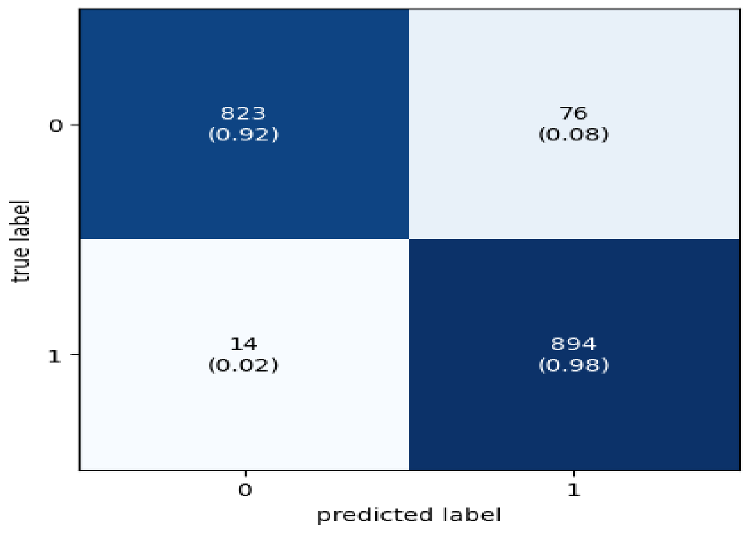


Figure 52 Confusion matrix for Spam-Ham (SMOTE) using GNB.

4.7.2.6 Spam-phishing (SMOTE) for SVC

Considering the spam-phishing dataset using the smote technique, the SVC classifiers was used to determine the classification of spam-phishing dataset, in the table 34 below, the classification report shows the precision is higher than recall and f1-score and the confusion matrix also confirms more precise predictions for the True positives.

	Precision	Recall	F1-Score	Support
0	0.90	0.99	0.94	129
1	0.99	0.89	0.94	132
Macro avg	0.95	0.94	0.94	261

Table 34 Classification Report for Spam-Phishing (SMOTE) SVC

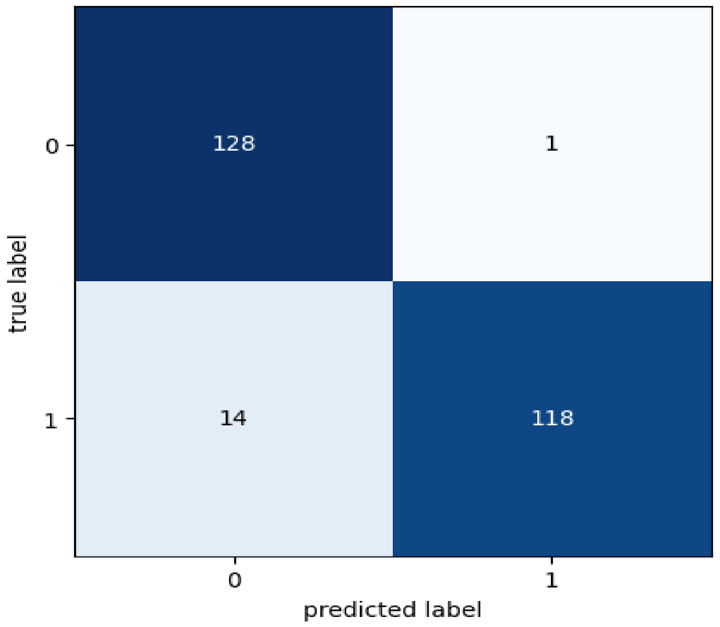


Figure 53 Confusion matrix for Spam-Phishing (SMOTE) using SVC.

4.7.2.7 Spam-phishing (SMOTE) for XGBoost

In spam-phishing dataset, XGBoost shows a significantly lower classification metrics compared to the spam-ham dataset with a macro average of 95%. The confusion matrix also shows the models categorizing into positives and negatives in figure 54 below.

	Precision	Recall	F1-Score	Support
0	0.94	0.95	0.95	129
1	0.95	0.94	0.95	132
Macro avg	0.95	0.95	0.95	261

Table 35 Classification Report for Spam-Phishing (SMOTE) XGBoost.

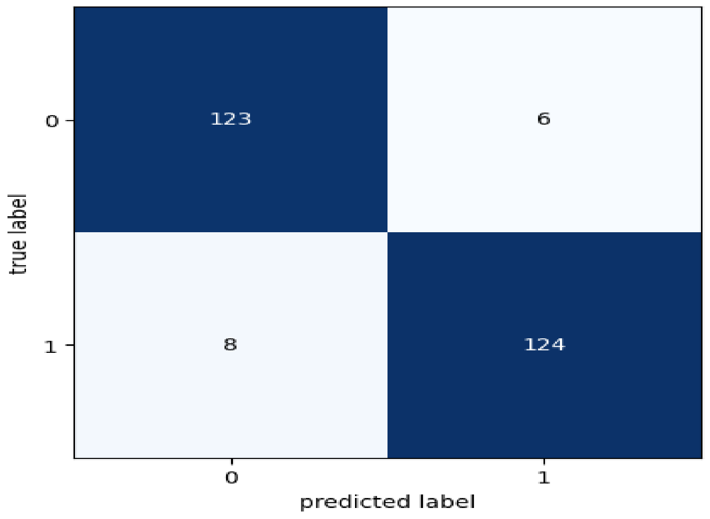


Figure 54 Confusion matrix for Spam-Phishing (SMOTE) using XGBoost.

4.7.2.8 Spam-phishing (SMOTE) for Multinomial NB

For spam-phishing dataset using MNB the model shows an improved classification report as compared against spam-ham data using the same classifier. The result shows the classification report and the confusion matrix in table 36 and figure 55 respectively.

	Precision	Recall	F1-Score	Support
0	0.92	0.97	0.94	129
1	0.97	0.92	0.94	132
Macro avg	0.94	0.94	0.94	261

Table 36 Classification Report for Spam-Phishing (SMOTE) MNB

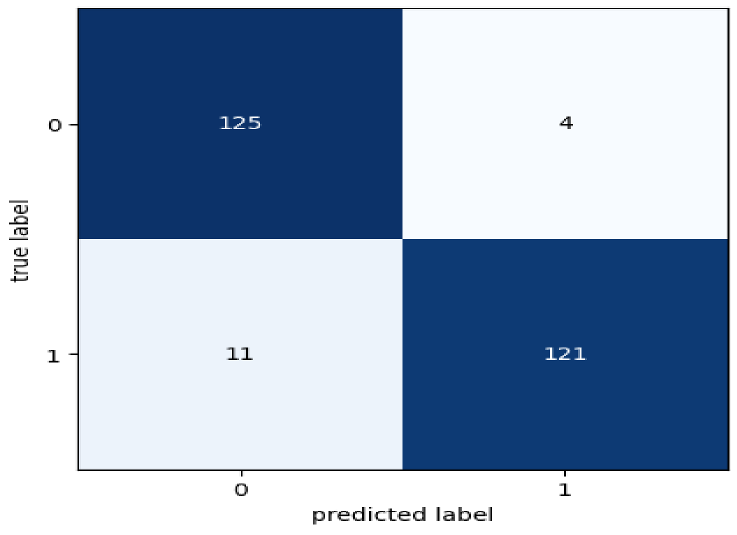


Figure 55 Confusion matrix for Spam-Phishing (SMOTE) using MNB.

4.7.2.9 Spam-phishing (SMOTE) for Random Forest Classifier

Using the Random Forest Classifier for Spam-phishing dataset the macro average shows a precision of 95% and a recall and f1-score of 94% each. The figure 56 shows the confusion matrix using the RF.

	Precision	Recall	F1-Score	Support
0	0.91	0.98	0.94	129
1	0.98	0.90	0.94	132
Macro avg	0.95	0.94	0.94	261

Table 37 Classification Report for Spam-Phishing (SMOTE) Random Forest.

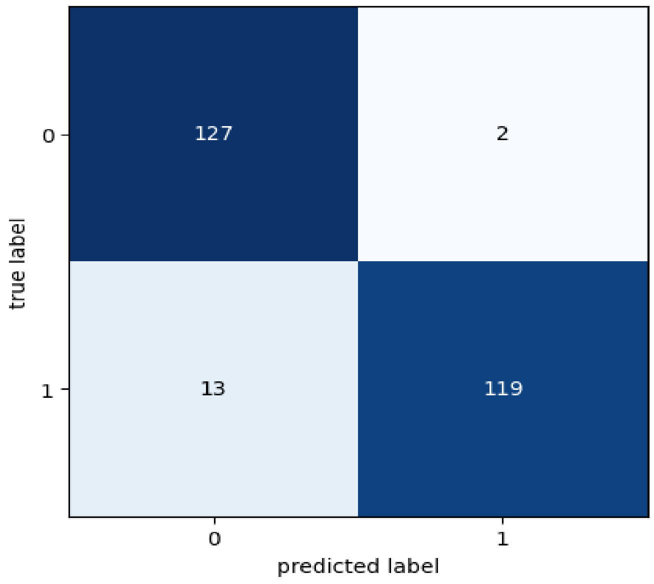


Figure 56 Confusion matrix for Spam-Phishing (SMOTE) using Random Forest.

4.7.2.10 Spam-phishing (SMOTE) for Gaussian NB

Gaussian was effective in classifying spam-phishing dataset using the smote technique, the result shows the macro average for precision, recall and f1-score to be 96% each. The confusion matrix for this class is illustrated below.

	Precision	Recall	F1-Score	Support
0	0.98	0.95	0.96	129
1	0.95	0.98	0.96	132
Macro avg	0.96	0.96	0.96	261

Table 38 Classification Report for Spam-Phishing (SMOTE) GNB

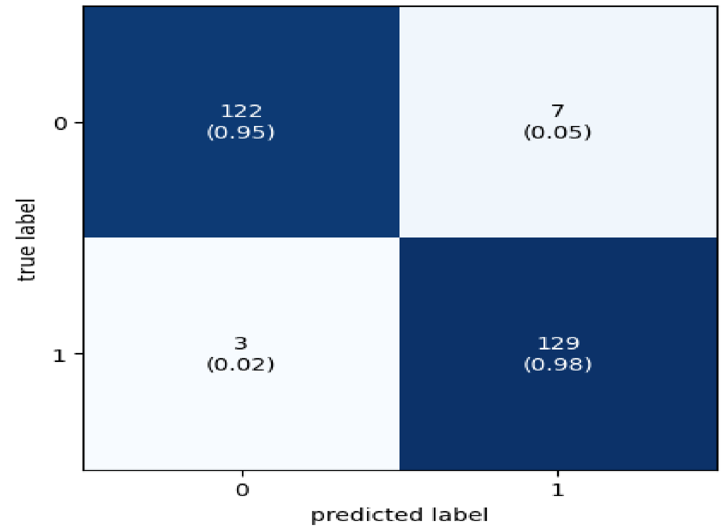


Figure 57 Confusion matrix for Spam-Phishing (SMOTE) using GNB.

4.7.2.11 Ham-phishing (SMOTE) for SVC

Considering a Ham-phishing dataset using the smote technique, the SVC classifier shows a very effective model for classifying the dataset. With a precision, recall and f1-score value of 97%. The result shows that SVC is very effective for this dataset with the confusion matrix illustrated below.

	Precision	Recall	F1-Score	Support
0	0.95	0.99	0.97	897
1	0.99	0.95	0.97	910
Macro avg	0.97	0.97	0.97	1807

Table 39 Classification Report for Ham-Phishing (SMOTE) for SVC

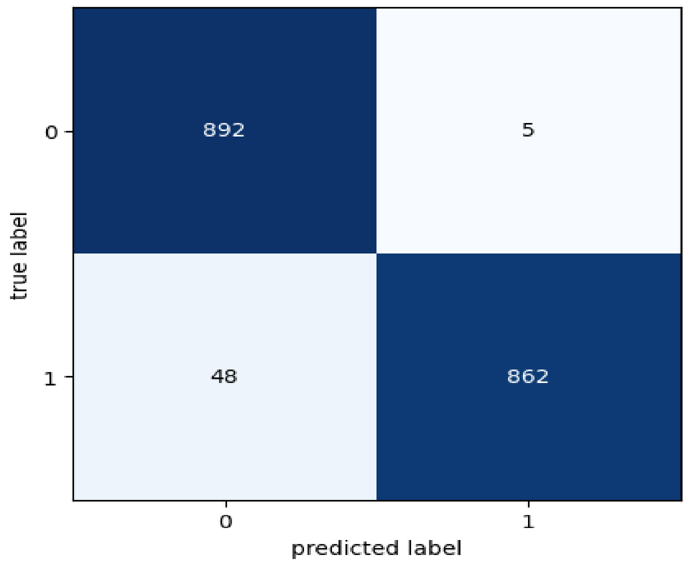


Figure 58 Confusion matrix for Ham-Phishing (SMOTE) using SVC.

4.7.2.12 Ham-phishing (SMOTE) for XGBoost.

In this model category the result best algorithm for classifying ham-phishing dataset using the smote technique. In the table 40 below the precision, recall and f1-score all had a macro average of 99%. Was a high confusion matrix class for the TP and TN.

	Precision	Recall	F1-Score	Support
0	1.00	0.98	0.99	897
1	0.98	1.00	0.99	910
Macro avg	0.99	0.99	0.99	1807

Table 40 Classification Report for Ham-phishing (SMOTE) XGBoost.

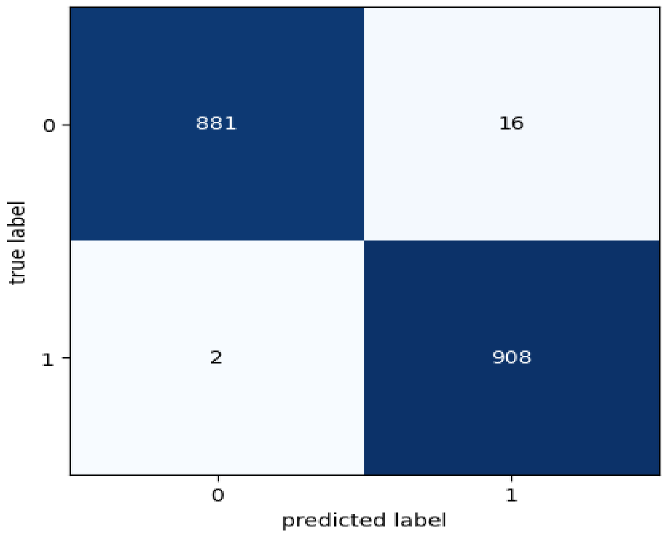


Figure 59 Confusion matrix for Ham-Phishing (SMOTE) using XGBoost.

4.7.2.13 Ham-phishing (SMOTE) for Multinomial NB

MNB algorithm for ham-phishing dataset in the table 41 below shows the precision, recall and f1 score has a macro average of 98%. This also shows that the model was effective in classifying the result, and the confusion matrix is illustrated in fig 60 below.

	Precision	Recall	F1-Score	Support
0	0.97	0.98	0.98	897
1	0.98	0.97	0.98	910
Macro avg	0.98	0.98	0.98	1807

Table 41 Classification Report for Ham-Phishing (SMOTE) MNB

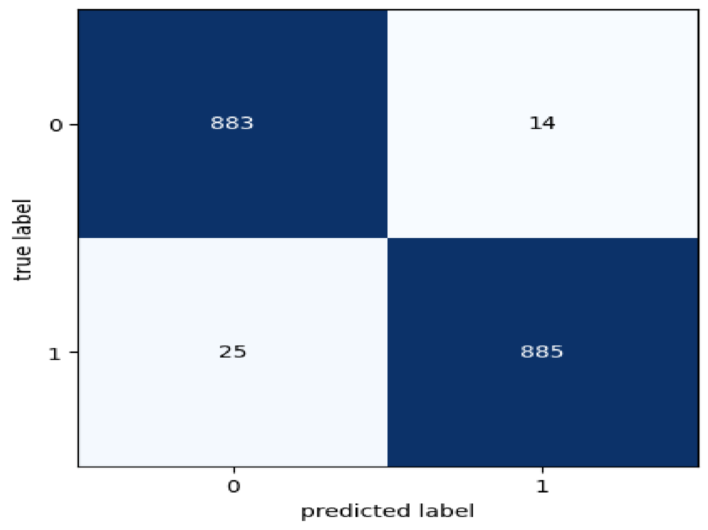


Figure 60 Confusion matrix for Ham-Phishing (SMOTE) using MNB.

4.7.2.14 Ham-phishing (SMOTE) for Random Forest Classifier

With ham having a greater percentage of this dataset the Random Forest Classifier was effective at accurately classifying the dataset, the table 42 shows the micro average of this classifier as 98%. With a confusion matrix illustrating high values for TP and TN.

	Precision	Recall	F1-Score	Support
0	0.96	1.00	0.98	897
1	1.00	0.95	0.97	910
Macro avg	0.98	0.98	0.98	1807

Table 42 Classification Report for Ham-Phishing (SMOTE) Random Forest.

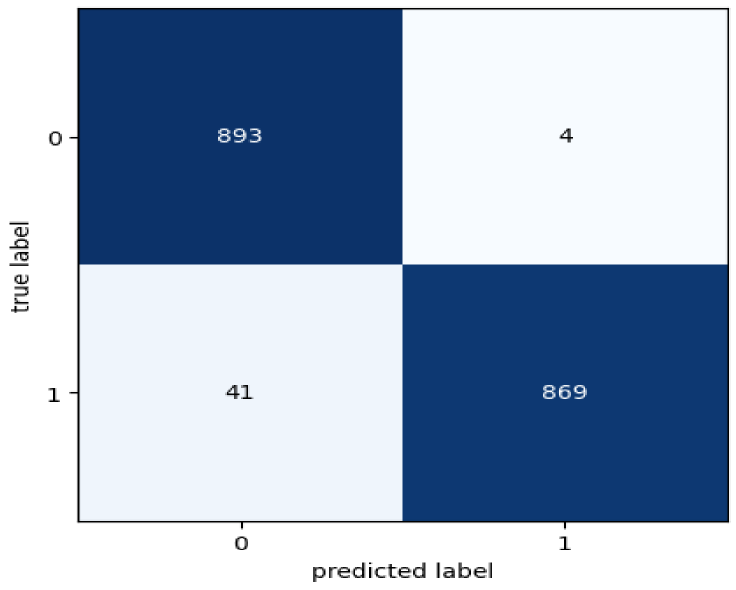


Figure 61 Confusion matrix for Ham-Phishing (SMOTE) using Random Forest

4.7.2.15 Ham-phishing (SMOTE) for Gaussian NB

GNM for Ham-phishing dataset using SMOTE, shows the precision, recall and f1 value as 98%. The confusion matrix also illustrates that 99% of the dataset has a class of 887 and 98% of the TN has a class of 891. The result is illustrated in figure 62 below.

	Precision	Recall	F1-Score	Support
0	0.98	0.99	0.98	897
1	0.99	0.98	0.98	910
Macro avg	0.98	0.98	0.98	1807

Table 43 Classification Report for Ham-Phishing (SMOTE) GNB

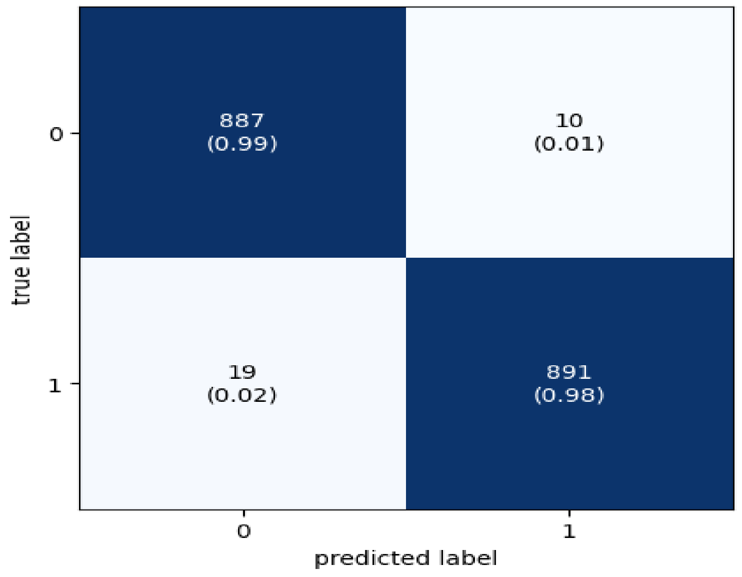


Figure 62 Confusion matrix for Ham-Phishing (SMOTE) using GNB.

4.7.3 Section 3

This section presents a comparison table for different classes of emails, showing the outcomes of machine learning algorithms. The table includes summaries for each machine learning algorithm, considering both SMOTE and non-SMOTE techniques.

4.7.3.1 Comparison between the machine learning algorithms.

The evaluation of machine learning algorithms using both SMOTE and non-SMOTE preprocessed data involved categorizing the data into three groups (spam-ham, spam-phishing, ham-phishing), and the results revealed that the SMOTE technique consistently demonstrated enhanced performance across all dataset categories, improving accuracy, precision, recall, and F1-score, making it a recommended approach for effectively addressing imbalanced datasets, particularly in the context of email classification.

4.7.3.1.1 Spam-Ham

Metric	SVC		Gaussian NB		XGBoost		Multinomial NB		Random Forest	
	With SMOTE	Without SMOTE	With SMOTE	Without SMOTE	With SMOTE	Without SMOTE	With SMOTE	Without SMOTE	With SMOTE	Without SMOTE

		E		E		E		E		
Accuracy	0.96	0.91	0.95	0.42	0.99	0.89	0.88	0.89	0.96	0.91
Precision	0.94	0.85	0.92	0.17	1.00	0.85	0.98	0.97	0.95	0.85
Recall	0.98	0.35	0.98	0.91	0.98	0.22	0.78	0.19	0.97	0.35
F1-Score	0.96	0.50	0.95	0.29	0.99	0.35	0.87	0.32	0.96	0.50

Table 44 Comparison between Smote and Non-Smote Performance Index for Spam-Ham.

4.7.3.1.2 Spam-Phishing.

Metric	SVC		Gaussian NB		XGBoost		Multinomial NB		Random Forest	
	With SMOTE	Without SMOTE	With SMOTE	Without SMOTE	With SMOTE	Without SMOTE	With SMOTE	Without SMOTE	With SMOTE	Without SMOTE
Accuracy	0.94	0.82	0.96	0.55	0.95	0.81	0.94	0.82	0.94	0.85
Precision	0.90	0.75	0.98	0.29	0.94	0.50	0.92	0.75	0.91	0.91
Recall	0.99	0.06	0.95	0.96	0.95	0.02	0.97	0.06	0.98	0.21
F1-Score	0.94	0.12	0.96	0.44	0.95	0.04	0.94	0.12	0.94	0.34

Table 45 Comparison between Smote and Non-Smote Performance Index for Spam-Phishing.

4.7.3.1.3 Ham-Phishing

Metric	SVC		Gaussian NB		XGBoost		Multinomial NB		Random Forest	
	With SMOTE	Without SMOT	With SMOTE	Without SMOT	With SMOTE	Without SMOT	With SMOTE	Without SMOT	With SMOTE	Without SMO

		E		E		E		E		TE
Accuracy	0.97	0.96	0.98	0.42	0.99	0.96	0.98	0.96	0.98	0.97
Precision	0.95	0.75	0.98	0.06	1.00	0.00	0.97	0.00	0.96	0.83
Recall	0.99	0.11	0.99	0.96	0.98	0.00	0.98	0.00	1.00	0.26
F1-Score	0.97	0.18	0.98	0.12	0.99	0.00	0.98	0.00	0.98	0.40

Table 46 Comparison between Smote and Non-Smote Performance Index for Ham-phishing.

4.8 Conclusion

Spam and Phishing emails are the most crucial in social networks, many issues arise through emails such as cost of dealing with spam and phishing emails due to their large quantities, privacy resulting in loss of sensitive information, time taken to identify spam and phishing emails, and cyber security threat due to malicious content. Using a spam and phishing detection approach, the model can quickly recognize spam and phishing emails and classify them before they become a threat to the organization. In this study, a machine learning and Natural Language processing-based supervised learning approach was used, and this plays a significant role in improving email classification. The dataset was prepared and dynamically classified into 3 categories namely spam-ham, spam-phishing, and ham-phishing. Different methods for effective classification were done on the dataset such as data preprocessing, feature selection, model training, model testing, classification result and performance evaluation. There were 5 machine learning algorithms used, and the result was evaluated using 8 performance indexes. In table 44 using the Spam-Ham dataset, the performance of each of the ML algorithms improves with the use of SMOTE, XGBoost shows the best performance, with an accuracy of 99%, precision of 100% recall of 98% and f1score of 99%. Table 45 likewise shows the comparison between SMOTE and Non-SMOTE for Spam-phishing dataset. GNB performed better compared to other ML algorithms with an accuracy of 96%, precision of 98%, recall of 95%, and f1score of 96%. In table 46 above the XGBoost outperformed the other ML algorithms compared for the Ham-Phishing dataset with an accuracy of 99%, precision of 100%, recall of 98%, and f1score of 99%. Hence the result shows that XGBoost machine learning algorithms generally outperformed other algorithms using the datasets. This research would help to improve categorizing emails into different folders based on their content, intent, or relevance and improve user experience and better manage email inboxes by automatically filtering, sorting and prioritizing messages.

CHAPTER 5

5 STUDY 3 – LOAN PREDICTION USING AZURE MACHINE LEARNING

5.1 Introduction

In the world of finance, credit risk poses a significant concern, representing the likelihood of financial loss arising from a borrower's inability to fulfill their loan obligations. Traditionally, credit risk has been mitigated by scrutinizing various borrower-related factors, such as their debt burden and income levels. In a pursuit to enhance and systematize the prediction of loan defaults, this research explores the effectiveness of machine learning, harnessing the power of the Azure cloud environment within the machine learning studio, employing designer components, using a comprehensive experiment approach. This multifaceted investigation involves a series of crucial steps, commencing with data cleaning and normalization, which ensures that the dataset is in a suitable form for analysis. Additionally, a sophisticated technique known as Synthetic Minority Over-sampling Technique (SMOTE) is utilized to address the class imbalance inherent in credit risk data. Furthermore, feature selection is employed to refine and optimize the dataset, streamlining it for use with machine learning algorithms. To classify loans effectively, the research utilizes supervised binary classification algorithms. Notable among them are the two-class decision forest, two-class support vector classifier (SVC), and two-class neural network. Each of these algorithms plays a distinct role in determining the loan status label, a crucial aspect of credit risk analysis. In the assessment phase, the research rigorously evaluates the results, employing key performance metrics such as accuracy, precision, recall, F1 score, and confusion matrices. Furthermore, the graphical representations of ROC curves, lift curves, and precision-recall relationships provide additional insights into the predictive power of the models. The culmination of this endeavor reveals that the decision forest algorithm stands out as the most effective classification algorithm for this dataset. The result shows the Two-class decision forest algorithm having an accuracy of 93.60%, precision of 94.10%, recall of 93.70% and f1 score of 93.91%. This research serves as confirmation of the power of machine learning in enhancing the predictability of credit risk, a vital aspect of financial risk management using azure environment for building models and deployment.

5.2 Background

Financial risk has been a huge problem in the finance industry, as the rate of technology keeps increasing, there have been various introduction of online lending platforms where customers can borrow money without necessarily visiting the bank. Hence banks have found it imperative to effectively determine ways to reduce risk through good decision-making to determine defaulter and non-defaulting customers. This research provides ways to improve financial risk management using machine learning techniques in Azure. It introduces different key concepts involving how the dataset was collected and gives details on the entire data flow to achieve the desired results. The effect of using this technique would significantly reduce bad debt (non-performing loans) hence increasing profitability and sustainability of financial institutions.

5.2.1 System Architecture

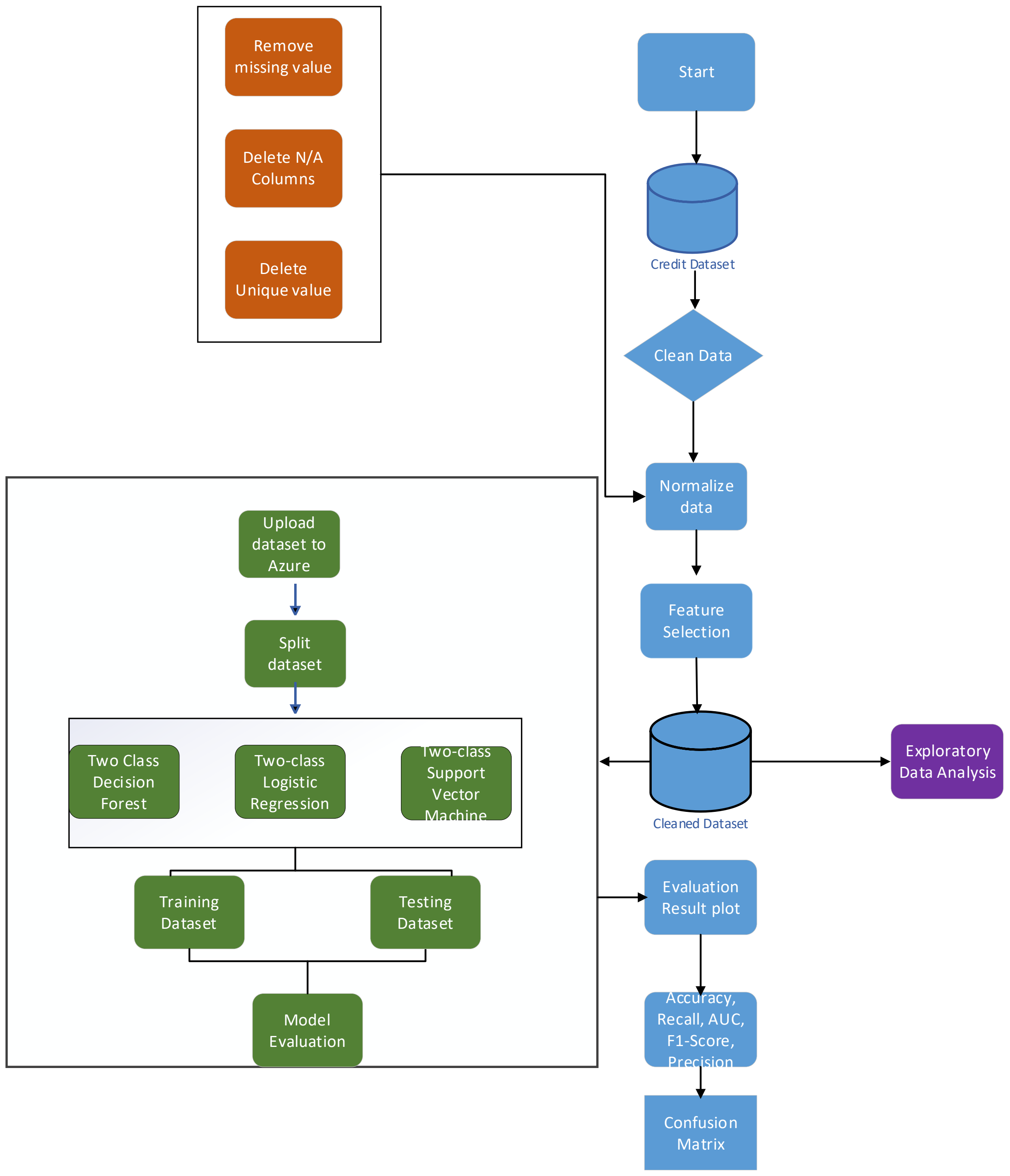


Figure 63 Proposed Methodology for Loan Prediction using Azure Machine Learning ((Alshouiliy et al., 2020))

5.2.2 Cloud computing

Cloud computing is a transformative IT model with three core service models: IaaS, PaaS, and SaaS. IaaS offers virtualized resources, giving control over the OS and apps. PaaS offers a complete development and deployment platform. SaaS delivers fully functional applications remotely. Key benefits include scalability for resource management, cost-efficiency by paying for actual usage, flexibility for remote access, reliability through redundancy, robust security measures, and access to cutting-edge technologies. Cloud use cases span data storage and backup, web hosting, big data analytics, machine learning, and IoT data management. Deployment models include public cloud for cost-effectiveness, private cloud for enhanced control, hybrid cloud for flexibility, and multi-cloud for diversity. Cloud computing revolutionizes IT by efficiently delivering resources over the internet, facilitating rapid scalability, and fostering innovation while reducing costs and administrative burdens.

5.2.3 Azure machine learning

Azure machine learning studio provides the ability to quickly create meaningful learning experiments and evaluate them for accuracy to enable them to be usable for prediction models. It consists of a series of steps which involve importing the dataset, creating a model, evaluating the model, refining and evaluating the model, deploying the model, test and use the model. There are different ML algorithms that can be performed on the ML studio such as classification, regression, and clustering. The system offers a comprehensive development, testing, and production environment for swiftly developing predictive analytic solutions.(Barnes, 2015).

5.2.4 Dataset

Data exist in different formats generally categorized into structured and unstructured data. The process of collection involves acquiring, extracting, and storing volumes of data which are used for various analyses. Data could be extracted from either a primary or a secondary source, primary data is data collected directly from the result of an interview, questionnaire, and surveys. Meanwhile, secondary data consists of data collected from the primary source which can be reusable. In this research, the dataset used was from a secondary source containing structured data from Kaggle (www.kaggle.com). The dataset has 32,582 rows and 12 columns, 390,984 observations, and has a total file size of 1.763 MB in CSV format. The dataset displays various attributes for determining credit risk, including "person_age," "person_income," "person_home_ownership," "person_emp_length," "loan intent," "loan_grade," "loan_amount," "loan_int_rate," "loan_status," "loan_percent_income," "cb_person_default_on_file," and "cb_person_cred_hist_length.".

5.2.5 Data Cleaning

Data cleaning is a crucial step in the data preparation process in Azure Machine Learning. It involves identifying and rectifying issues in your dataset to ensure that it is of high quality and suitable for building machine learning models. It involves handling missing data, the columns are selected for cleaning using the clean component in azure machine learning designer.

5.2.6 Data Normalization

Data normalization is a fundamental step in preparing data for machine learning. It involves transforming numeric columns to a standard scale, typically [0.0, 1.0], to ensure uniformity across all records and fields. The purpose of normalization is to prevent varying scales from affecting model performance. It offers benefits like consistency, improved model accuracy, and efficient convergence. Common techniques include Min-Max Normalization, Z-score Normalization, and Normalization by Decimal Scaling. In summary, data normalization is essential for ensuring fair and accurate model training by standardizing feature scales.(varshachoudhary, 2023)

5.2.7 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical phase within the machine learning process, serving multiple essential functions. Its primary objective is to gain a deep understanding of a dataset's characteristics and patterns. It encompasses various tasks, including grasping the dataset's size, structure, and data types, which are foundational for effective analysis. It delivers descriptive statistics and measures like mean and median, offering insights into central tendencies and data variations, extending beyond just numerical summaries. Furthermore, EDA employs data visualization techniques, like charts and graphs, to unveil patterns, relationships, and outliers, providing valuable insights that go beyond numeric descriptions. Identifying outliers is a key element, as outliers can significantly impact analysis outcomes and necessitate appropriate handling. It also extends to spotting trends, patterns, and correlations within the data, facilitating data-driven decision-making. EDA aids in effective communication by using visualizations and summaries to convey findings to both technical and non-technical audiences. (Hartwig & Dearing, 1979)

5.2.8 Features selection

In Azure Machine Learning Studio, feature selection is a crucial data preprocessing step that involves selecting relevant features while excluding irrelevant ones from a dataset. For instance, in customer loan prediction, selecting features like customer age, usage frequency, and customer support interactions, while excluding less informative variables like customer ID. Azure ML Studio provides various methods like filter-based, wrapper-based, and embedded feature selection. In a credit risk assessment model, including employment history while excluding irrelevant factors like email addresses. Evaluate the impact of feature selection by comparing model performance metrics before and after. Experimentation and iteration may lead to improved accuracy, such as selecting the most influential features for a loan prediction model. Integrating feature selection seamlessly into the machine learning pipeline, ensured that only the most relevant features are used for model training, ultimately enhancing efficiency and accuracy. (Kumar & Minz, 2014)

5.2.9 Split Data

The size of the datasets and the training/testing split ratios greatly affect the outcome of the models and the model performance during a classification algorithm of the machine learning process. Several machine learning models was compared using the split ratios of the dataset. Significant differences could be detected between the train/test split hence determining the result of the test validation (Rácz et al., 2021). The dataset was divided into train and test datasets with 70% training data and 30% testing.

5.2.10 ML Algorithms

5.2.10.1 Two-class decision forest

This is a machine learning model used for binary classification tasks, it can also be referred to as binary decision forest. The ensemble learning method is based on decision trees, where data is classified into two possible classes or categories. It works based on two sampling methods, the replicate method trains each tree on the same training data whereas the bootstrap aggregates each tree on a new sample(Rajagopal et al., 2020). In Azure, the two classes are added to the designer component among the Machine Learning Algorithms.

5.2.10.2 Two class Neural Networks

The neural network approach is utilized in this module for binary classification. A neural network is a collection of interconnected layers that is used to address a wide range of difficult AI issues. Because of their non-linearity, variable interactions, and customization benefits, they frequently outperform conventional machine learning models. This algorithm builds a network of input, output, and hidden layers for the data used to forecast the target class.(Singh, 2020)

5.2.10.3 Two class Support Vector Machine

An SVM is a machine learning algorithm used for classification tasks, its finds a hyperplane that maximizes the margin between two classes in a binary classification problem (Shivanna & Agrawal, 2020). The hyperplanes give the decision boundary separating data points from different classes, it also provides a maximum safety margin or separation between the classes. It classifies datapoints into one of two distinct categories namely Yes and No.

5.2.11 Score Model

The model score is an Azure component added to the designer pipeline after training. The data to be scored should be in a format compatible with the trained model been used.

5.2.12 Model Evaluation

This is an Azure component that measures the performance of the trained model. The dataset containing the scored model has been evaluated using a set of industry-standard evaluation metrics. Evaluation of the model can be performed on different types of ML algorithms used such as Classification models, Regression Models and Clustering model.

5.3 Methodology

The methodology used in this research involves predicting bank credit defaults in the financial sector using machine learning on Azure cloud. The azure machine learning platform is a cloud-based platform designed to provide the entire machine learning lifecycle involving data preparation, model training, deployment, and monitoring in the cloud. It provides scalability and flexibility, ease of use, integration, collaboration, cost management and security. The following steps show the methods used in predicting credit defaulters in the financial sector.

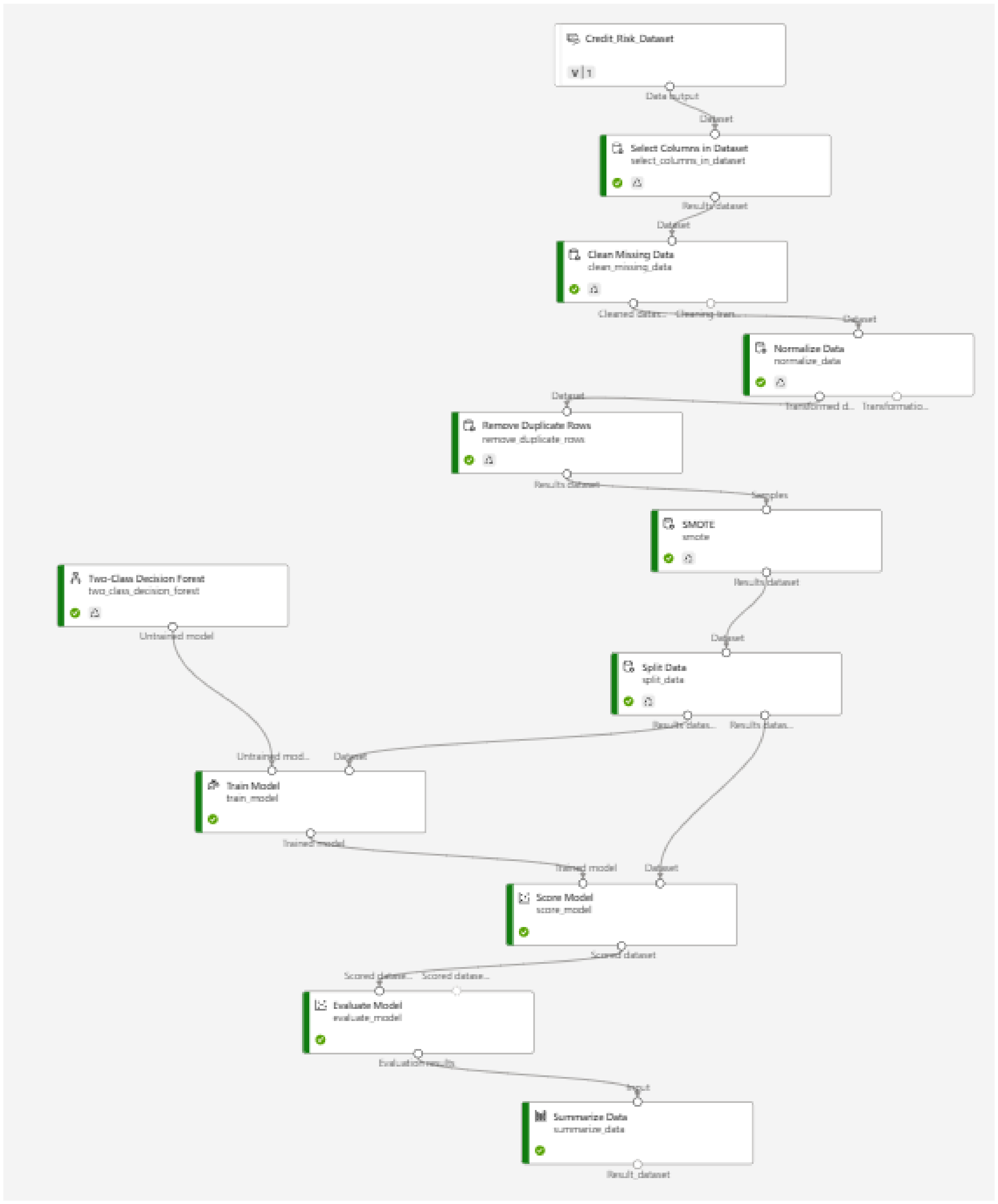
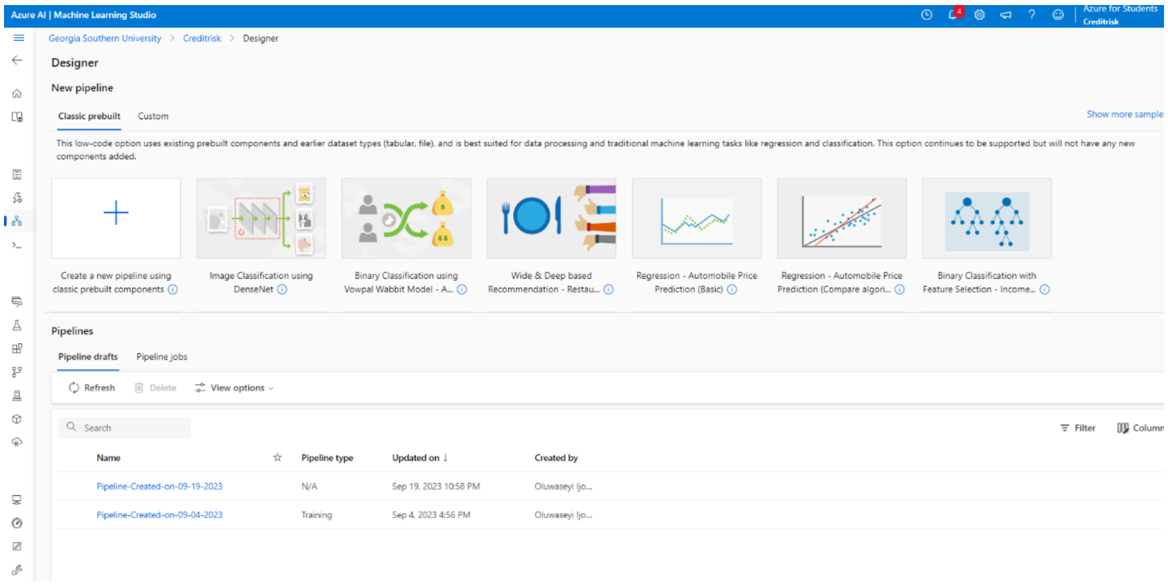


Figure 64 Azure Machine learning Component

5.3.1 Azure Machine Learning Designer

This is an Azure machine leaning platform in the Azure studio that allows users to create machine learning based project on the cloud using the drag and drop interface. The dataset is first imported into the pipeline in the data input and output category of the designer.


Azure AI | Machine Learning Studio

Azure for Students
Creditrisk

Georgia Southern University > Creditrisk > Designer

Designer

New pipeline

Classic prebuilt | Custom | Show more sample

This low-code option uses existing prebuilt components and earlier dataset types (tabular, file), and is best suited for data processing and traditional machine learning tasks like regression and classification. This option continues to be supported but will not have any new components added.

- Create a new pipeline using classic prebuilt components
- Image Classification using DenseNet
- Binary Classification using Vowpal Wabbit Model - A...
- Wide & Deep based Recommendation - Restau...
- Regression - Automobile Price Prediction (Basic)
- Regression - Automobile Price Prediction (Compare algori...
- Binary Classification with Feature Selection - Income...

Pipelines

Pipeline drafts | Pipeline jobs

Refresh | Delete | View options

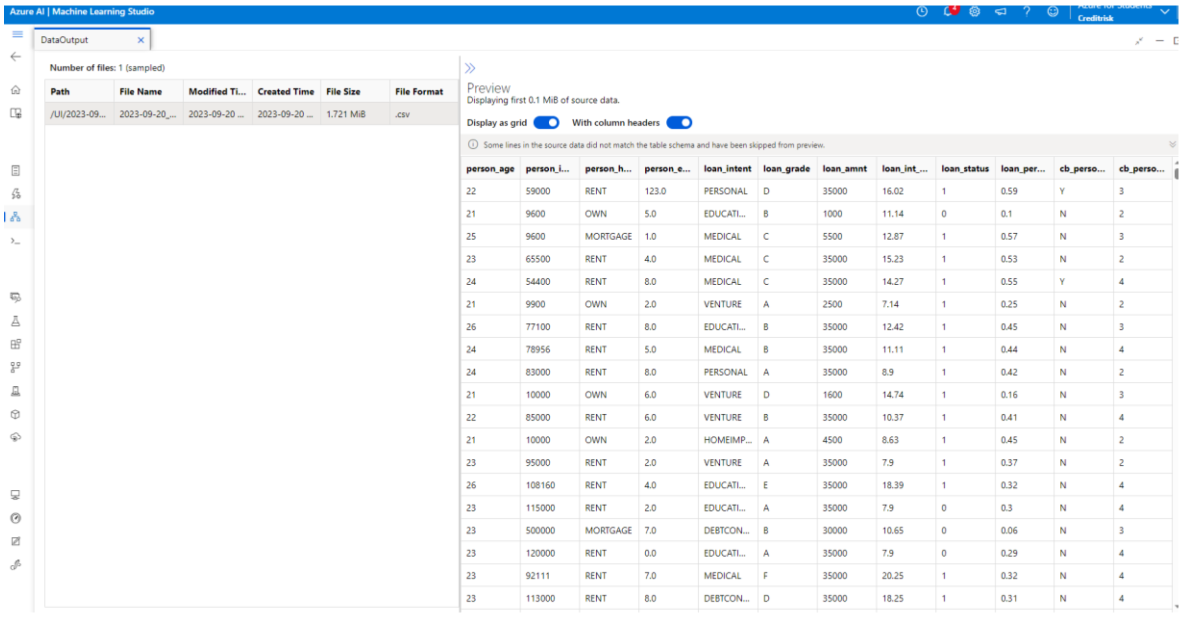
Search | Filter | Column

Name	Pipeline type	Updated on ↓	Created by
Pipeline-Created-on-09-19-2023	N/A	Sep 19, 2023 10:58 PM	Oluwaseyi Ijo...
Pipeline-Created-on-09-04-2023	Training	Sep 4, 2023 4:56 PM	Oluwaseyi Ijo...

Figure 65 Azure Machine Learning Studio Environment

5.3.2 Dataset

The dataset consists of a credit risk dataset downloaded from Kaggle which consists of 390,984 observations, with 32,582 rows and 12 columns with a total file size of 1.763MB in a CSV format. The dataset shows different attributes for credit risk determination such as “person_age”, “person_income”, “person_home_ownership”, “person_emp_length”, “loan intent”, “loan_grade”, “loan_amount”, “loan_int_rate”, “loan_status”, “loan_percent_income”, “cb_person_default_on_file”, “cb_person_cred_hist_length”. The target class called the loan_status is encoded with 0 and 1 with 0 which signifies customers non-default and 1 signifies default. The data was moved from the local device into the cloud storage resources while designing a pipeline to exchange data. In Azure there are different ways data could be imported into the ML studio such as the Azure Blob container, Azure File Share, Azure Data Lake, Azure Data Lake Gen2 Azure SQL Database and Azure PostgreSQL. The output of the data imported is a dataset that can be used with the designer pipeline.



Number of files: 1 (sampled)

Path	File Name	Modified Ti...	Created Time	File Size	File Format
/UI/2023-09...	2023-09-20_...	2023-09-20 ...	2023-09-20 ...	1.721 MiB	.csv

Preview
Displaying first 0.1 MiB of source data.

Display as grid | With column headers

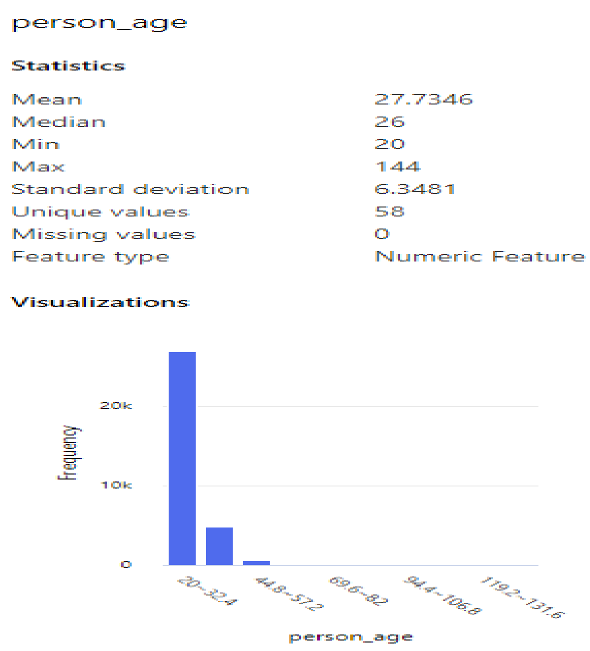
Some lines in the source data did not match the table schema and have been skipped from preview.

person_age	person_i...	person_h...	person_e...	loan_intent	loan_grade	loan_amnt	loan_int_...	loan_status	loan_per...	cb_perso...	cb_perso...
22	59000	RENT	123.0	PERSONAL	D	35000	16.02	1	0.59	Y	3
21	9600	OWN	5.0	EDUCATI...	B	1000	11.14	0	0.1	N	2
25	9600	MORTGAGE	1.0	MEDICAL	C	5500	12.87	1	0.57	N	3
23	65500	RENT	4.0	MEDICAL	C	35000	15.23	1	0.53	N	2
24	54400	RENT	8.0	MEDICAL	C	35000	14.27	1	0.55	Y	4
21	9900	OWN	2.0	VENTURE	A	2500	7.14	1	0.25	N	2
26	77100	RENT	8.0	EDUCATI...	B	35000	12.42	1	0.45	N	3
24	78956	RENT	5.0	MEDICAL	B	35000	11.11	1	0.44	N	4
24	83000	RENT	8.0	PERSONAL	A	35000	8.9	1	0.42	N	2
21	10000	OWN	6.0	VENTURE	D	1600	14.74	1	0.16	N	3
22	85000	RENT	6.0	VENTURE	B	35000	10.37	1	0.41	N	4
21	10000	OWN	2.0	HOMEIMP...	A	4500	8.63	1	0.45	N	2
23	95000	RENT	2.0	VENTURE	A	35000	7.9	1	0.37	N	2
26	108160	RENT	4.0	EDUCATI...	E	35000	18.39	1	0.32	N	4
23	115000	RENT	2.0	EDUCATI...	A	35000	7.9	0	0.3	N	4
23	500000	MORTGAGE	7.0	DEBTCON...	B	30000	10.65	0	0.06	N	3
23	120000	RENT	0.0	EDUCATI...	A	35000	7.9	0	0.29	N	4
23	92111	RENT	7.0	MEDICAL	F	35000	20.25	1	0.32	N	4
23	113000	RENT	8.0	DEBTCON...	D	35000	18.25	1	0.31	N	4

Figure 66 Dataset Import to Azure ML studio.

5.3.3 Explanatory data analysis

In this section the credit_risk dataset was further explored using data analysis on the Azure. Azure Machine Learning studio designer was used to draft a flowchart of executable instructions using the drag and drop options. By digging deep into the dataset explanatory data analysis gives more insight into the model we build. The summary statistics of each column of the dataset was calculated and the result is shown below. In the figure 67 below the person_age shows the age range of customer requesting for loan with their mean age of 27years. The data shows that most age ranges around 20-32 year of age.



person_age

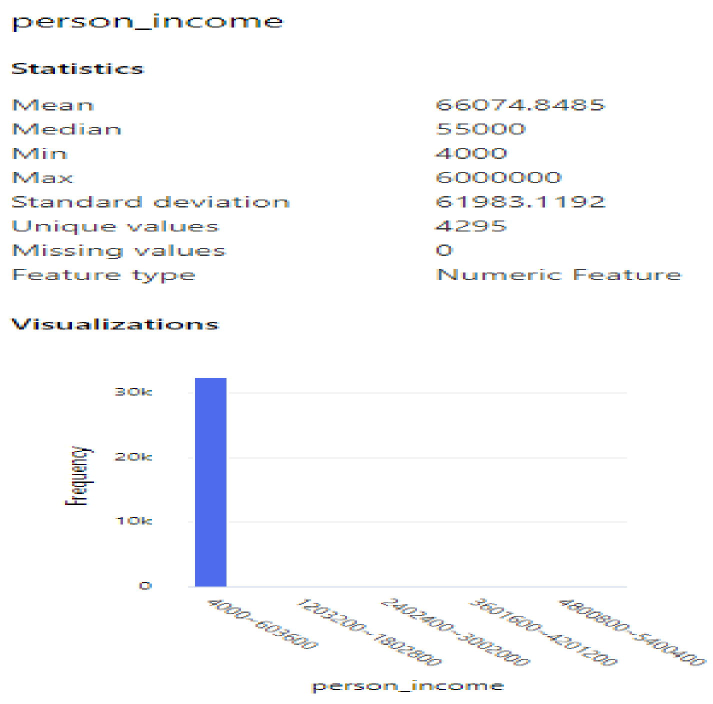
Statistics

Mean	27.7346
Median	26
Min	20
Max	144
Standard deviation	6.3481
Unique values	58
Missing values	0
Feature type	Numeric Feature

Visualizations

Figure 67 EDA and summary statistics for Person-age.

The descriptive statistics for person_income is also highlighted below. the result of the EDA shows that the average income of the bank customers requesting a loan facility is 6,6074, median of 55,000, minimum income as 4,000 and maximum income of 600,000. There were 4,295 unique data point and the total person_income amount ranges between 400- 603,600.



person_income

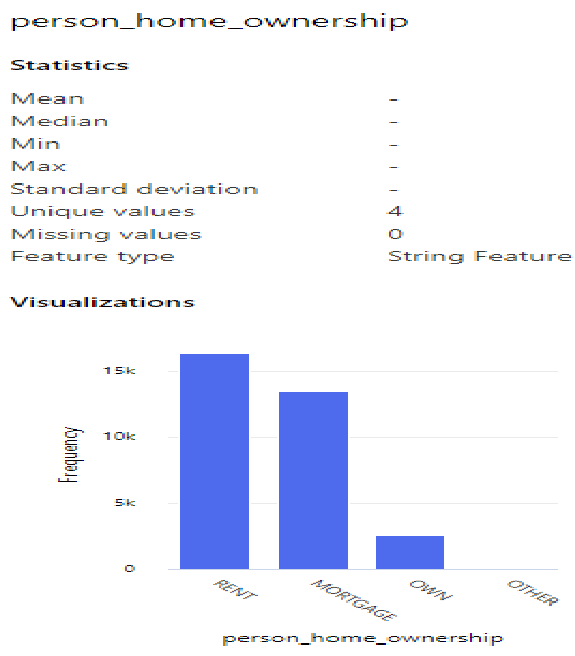
Statistics

Mean	66074.8485
Median	55000
Min	4000
Max	6000000
Standard deviation	61983.1192
Unique values	4295
Missing values	0
Feature type	Numeric Feature

Visualizations

Figure 68 EDA and summary statistics for Person-income

In the case of home_ownership, this visualization shows either the customer has a home, paying for rent or mortgage. From the visualization below, we can conclude that most of the customer spends more on rent fees and the tendency for collecting loan for rent payment is high. There are 4 unique values in this dataset with values ranging from rent, mortgage, own, and others.



person_home_ownership

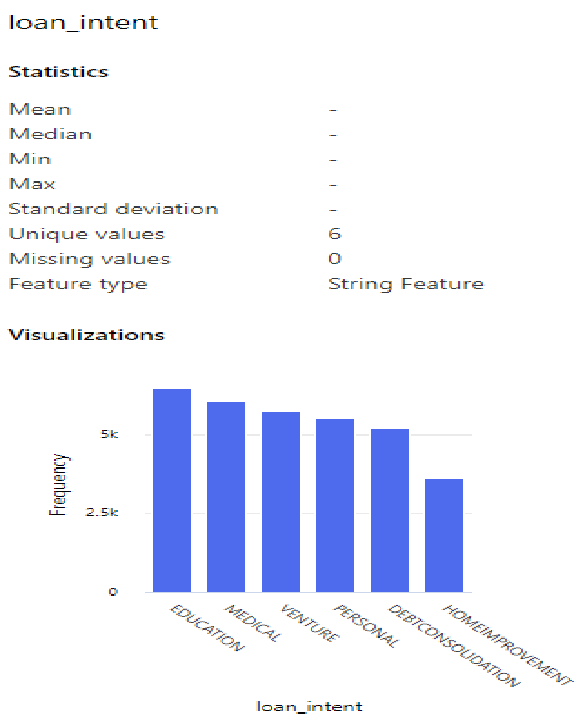
Statistics

Mean	-
Median	-
Min	-
Max	-
Standard deviation	-
Unique values	4
Missing values	0
Feature type	String Feature

Visualizations

Figure 69 EDA and summary statistics for person_home_ownership

Financial institutions need to understand the intent of customers' loan request in other to determine the probability of repayment. Visualizing the intent of customers loan request shows the different categories of why the loan is needed and gives insight to understand if such intent has the capacity to yield repayment. In the case of the figure below, the majority of the customers spends more on education loan compared to other loan intent such as medical, venture, personal, debit consolidation, and home improvement.



loan_intent

Statistics

Mean	-
Median	-
Min	-
Max	-
Standard deviation	-
Unique values	6
Missing values	0
Feature type	String Feature

Visualizations

Figure 70 EDA and summary statistics for Loan_intent

Visualizing the loan amount most customers receive from the bank indicates that the average amount of loan lenders gives to borrowers is 9,589 with the maximum loan amount to be 35,000 and the minimum loan amount to be 500. Majority of the loans issued to customers ranges between 7,400 and 10,850.

loan_amnt

Statistics

Mean	9589.3711
Median	8000
Min	500
Max	35000
Standard deviation	6322.0866
Unique values	753
Missing values	0
Feature type	Numeric Feature

Visualizations

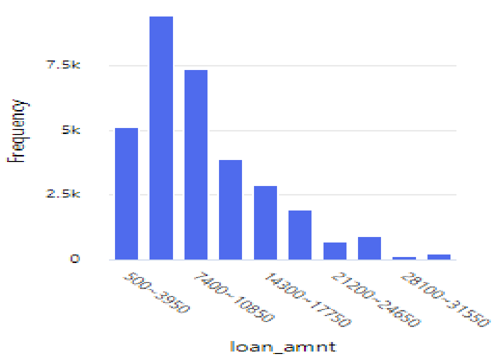
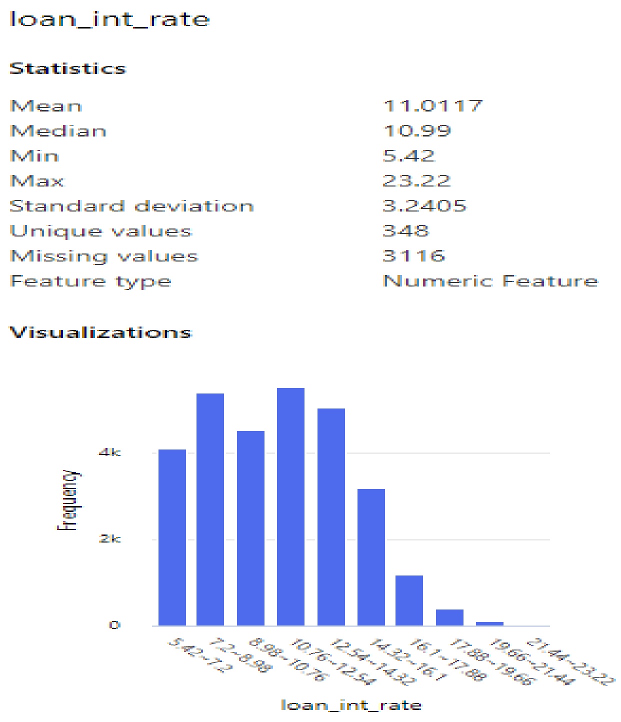


Figure 71 EDA and summary statistics for Loan_Amount

The main aim of every financial institution is to make profit, hence various interest rates are given to the customer based on the amount of loan disbursed. The average interest rate is 11%, with the minimum interest rate given is 5.4% and the maximum interest rate is 23.22%. from the result of the visualization below, the range with the highest interest rate occurrence is between 10.76-12.54 where most rate lies.



loan_int_rate

Statistics

Mean	11.0117
Median	10.99
Min	5.42
Max	23.22
Standard deviation	3.2405
Unique values	348
Missing values	3116
Feature type	Numeric Feature

Visualizations

Figure 72 EDA and summary statistics for Loan_interest_rate

Loan _percent_income demonstrates the percentage of the customer’s income to the value of the loan. The average percent_income is 17% of their income, with 15% been the lowest percent and 83% of customers income been the highest percent income. It is safe to say that customers with high loan_percent_income, have lower tendency of loan repayment compared to customers with low loan_percent_income.

loan_percent_income

Statistics

Mean	0.1702
Median	0.15
Min	0
Max	0.83
Standard deviation	0.1068
Unique values	77
Missing values	0
Feature type	Numeric Feature

Visualizations

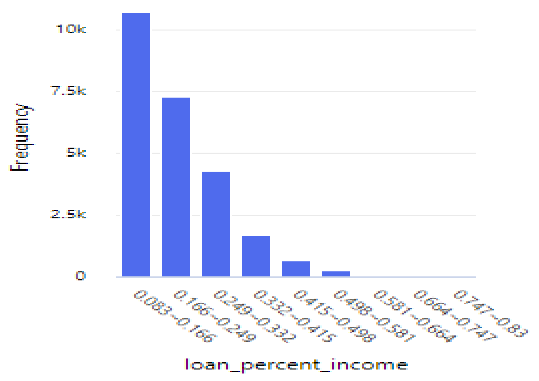
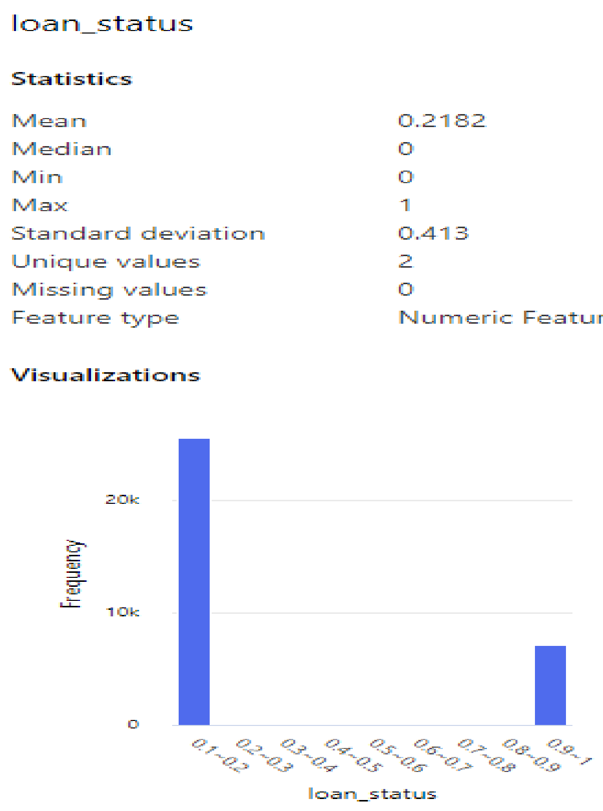


Figure 73 EDA and summary statistics for Loan_percent_income

The loan status in this example shows a visual representation of the imbalanced dataset with non-defaults identified by 0 and defaulters identified by 1. The visualization shows that the rate of defaulters appears as the minority class hence this research suggested the use of SMOTE technique.



loan_status

Statistics

Mean	0.2182
Median	0
Min	0
Max	1
Standard deviation	0.413
Unique values	2
Missing values	0
Feature type	Numeric Feature

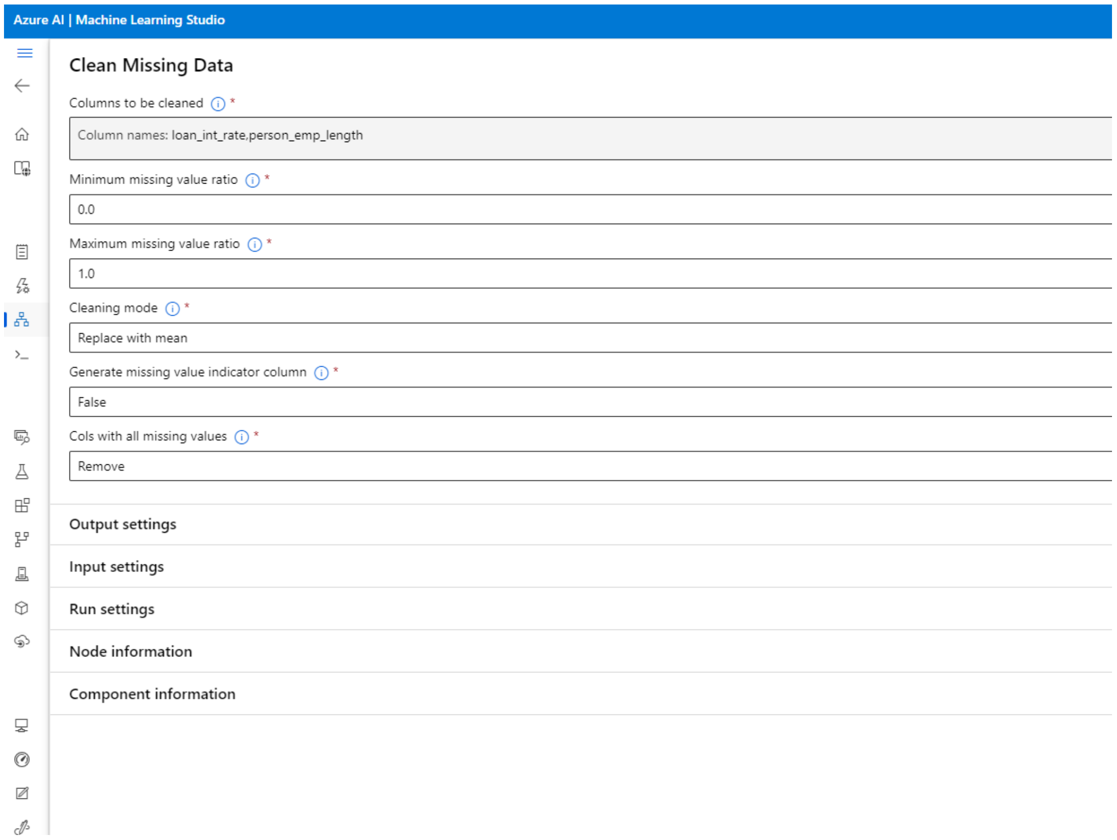
Visualizations

Figure 74 EDA and summary statistics for Loan_status

5.3.4 Data Transformation

5.3.4.1 Cleaning Missing Data

Dataset usually comes from different sources both in structured and unstructured format. The goal of a data cleaning operation is to prevent problems caused by missing data before training. This process involves different multiple types of operation including replacement of missing values, removal of rows and columns containing missing values and inferring values based on statistical methods. In Azure machine learning the sources of the dataset remains the same after cleaning creating a new dataset in the workspace for subsequent processing. The result of a data cleaning process is a transformed and clean data which are reuseable. There are several ways to clean up a dataset in Azure, including custom substitution values, mean, median, mode, remove full row, and remove complete column. In the dataset used the loan_interest_rate consist of missing column data of 3116 which was subsequently cleaned, 895 missing record was found in person_emp_length column.



Azure AI | Machine Learning Studio

Clean Missing Data

Columns to be cleaned *

Column names: loan_int_rate,person_emp_length

Minimum missing value ratio *

0.0

Maximum missing value ratio *

1.0

Cleaning mode *

Replace with mean

Generate missing value indicator column *

False

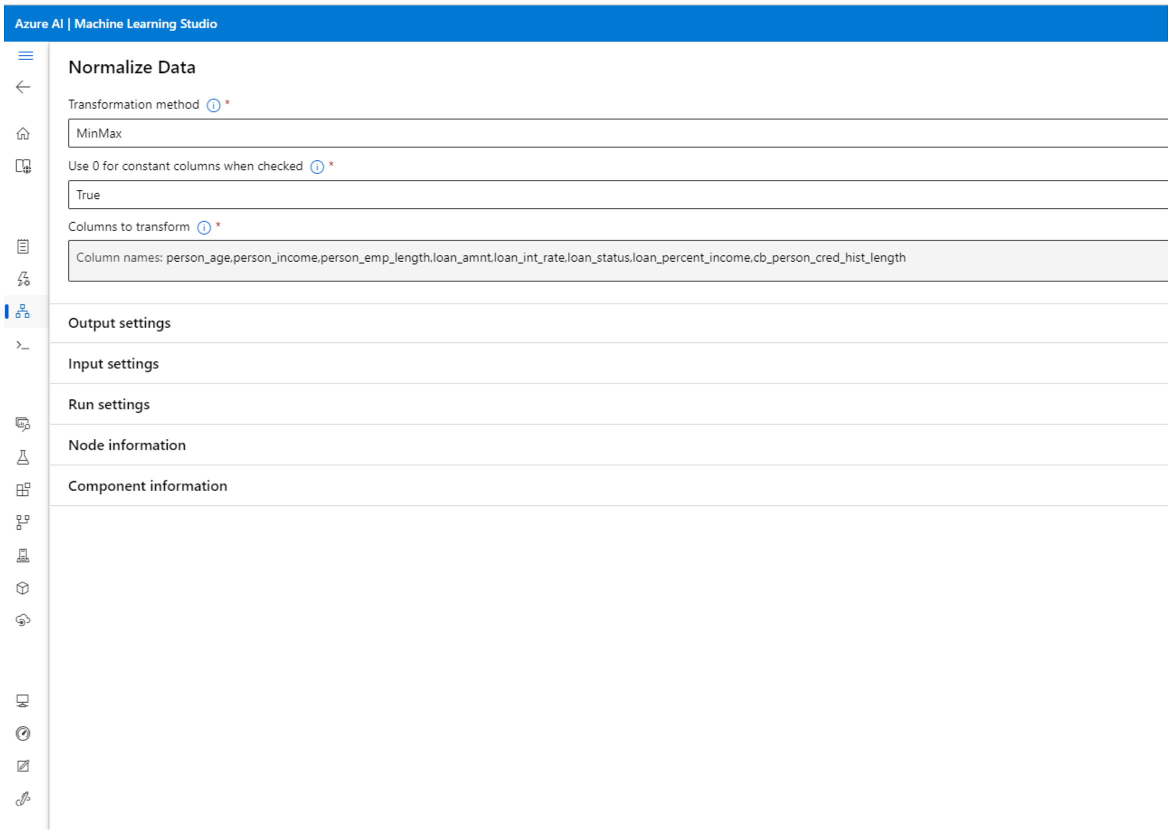
Cols with all missing values *

Remove

Figure 75 Data Cleaning using clean missing data component in Azure ML.

5.3.4.2 Normalize data.

Normalization is a data preparation technique used in machine learning, it is typically used to convert the values of numeric columns into a standard scale in a dataset without distorting the differences in ranges or losing vital information about the dataset. It avoids the problem of differences in scale of individual columns in the dataset by maintaining the overall distribution and keeping each value within a scale applied across all numeric columns. All values could be changed between 0 and 1. Normalization is configured in azure by adding the Normalize data component to the pipeline under the data transformation component. It is achieved by connecting a dataset that contains at least one column. Every feature was rescaled to the range [0, 1] using the MinMax normalizer. whereas 0 is the lowest value and 1 is the highest value.


Azure AI | Machine Learning Studio

Normalize Data

Transformation method *

MinMax

Use 0 for constant columns when checked *

True

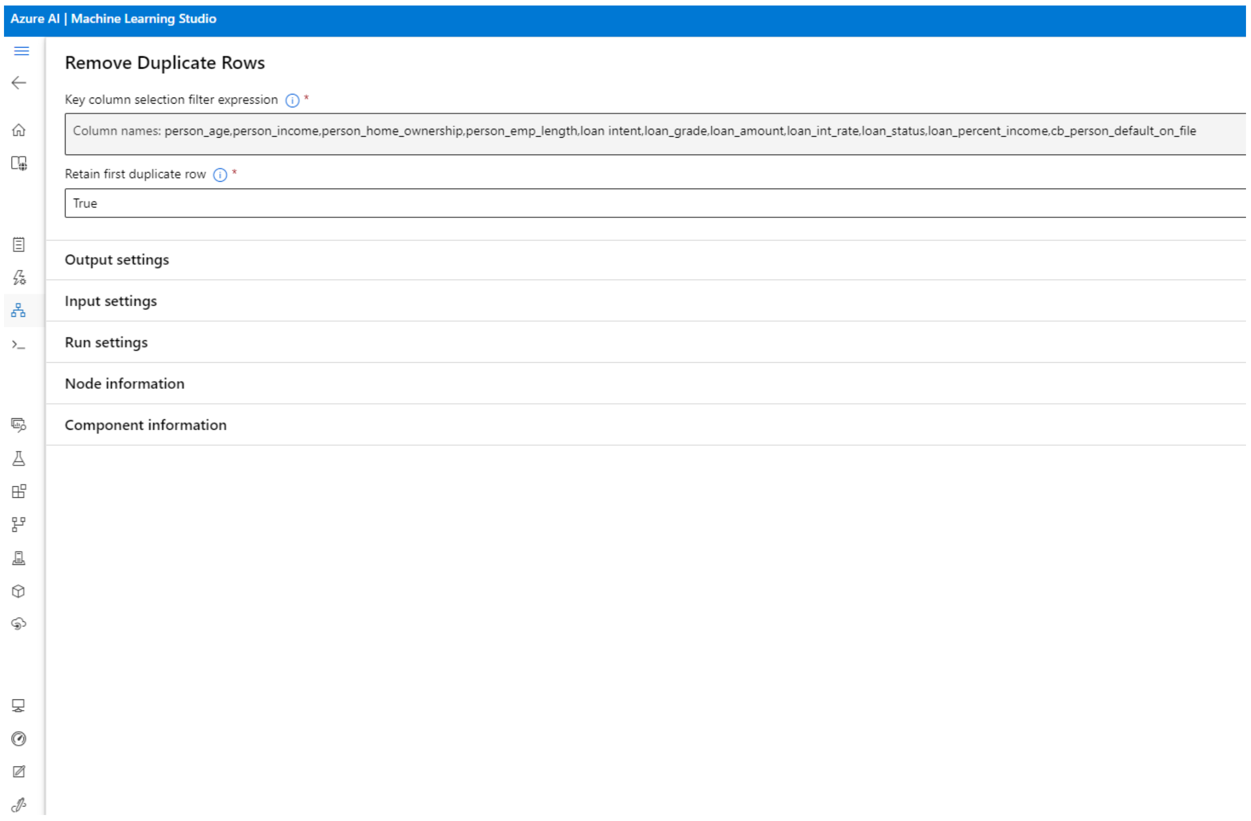
Columns to transform *

Column names: person_age,person_income,person_emp_length,loan_amnt,loan_int_rate,loan_status,loan_percent_income,cb_person_cred_hist_length

Figure 76 Data Normalization in Azure ML Studio.

5.3.4.3 Remove Duplicate data.

This component in Azure removes potential duplicates from the dataset. The component is added to the azure pipeline under data transformation. The input dataset is first connected to the duplicate rows. The first duplicate row which row to return when duplicates are found, the row therefore is returned and other are discarded.



Remove Duplicate Rows

Key column selection filter expression *

Column names: person_age,person_income,person_home_ownership,person_emp_length,loan intent,loan_grade,loan_amount,loan_int_rate,loan_status,loan_percent_income,cb_person_default_on_file

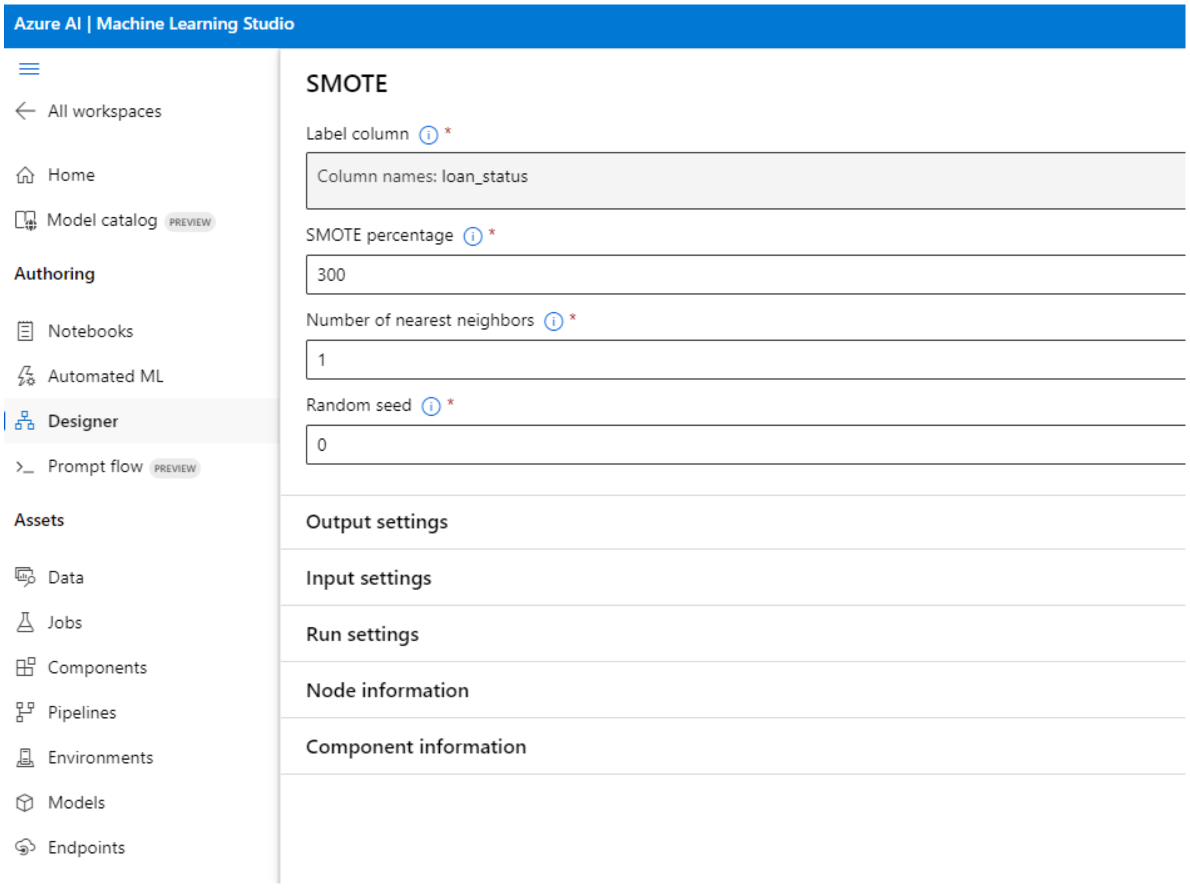
Retain first duplicate row *

True

Figure 77 Duplicate Remover using Azure ML Studio component.

5.3.4.4 Smote

The Synthetic Minority Oversampling Technique (SMOTE) is a vital tool in machine learning, specifically designed to tackle imbalanced datasets. In this dataset, which comprises 32,582 rows and columns, there is a significant class imbalance, with non-defaulters making up 78% and defaulters accounting for only 22% of the data. This class imbalance can lead to biased machine learning models. SMOTE, or Synthetic Minority Oversampling Technique, addresses this issue by focusing on the minority class, in this case, the defaulters. Its primary objective is to increase the representation of the minority class in the dataset, thus mitigating bias. In this research, the minority percentage choose was 300, signifying an increase in the number of minority class samples to 300% of their original count. In Azure Machine Learning, SMOTE is implemented as a data transformation component. Users can select the target column representing the minority class, often the "defaulters," and specify the desired oversampling percentage. By applying SMOTE, you effectively balance the dataset, ensuring a more equitable representation of both classes. This not only enhances the performance of machine learning models but also ensures fairness, especially in cases where imbalanced datasets can introduce bias into the analysis.



SMOTE

Label column *
Column names: loan_status

SMOTE percentage *
300

Number of nearest neighbors *
1

Random seed *
0

Figure 78 Data Preprocessing using SMOTE Component

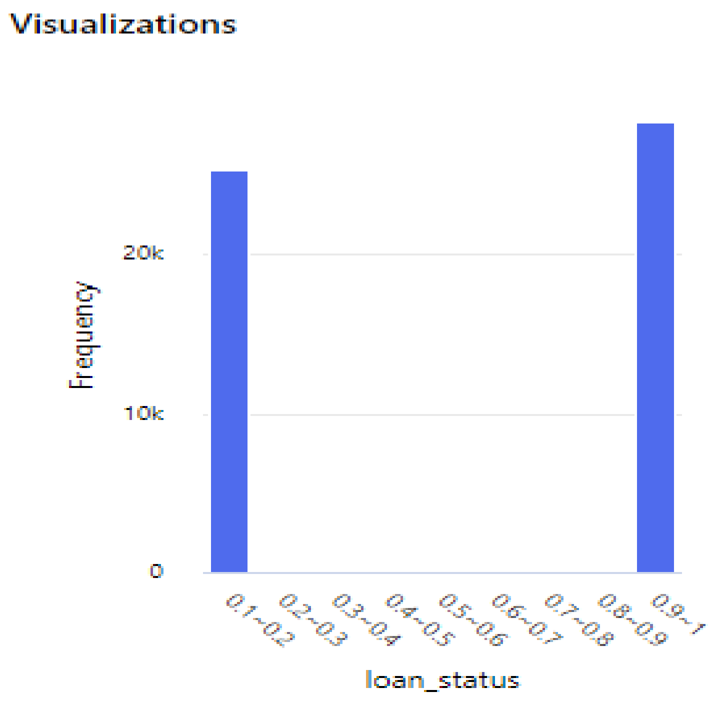
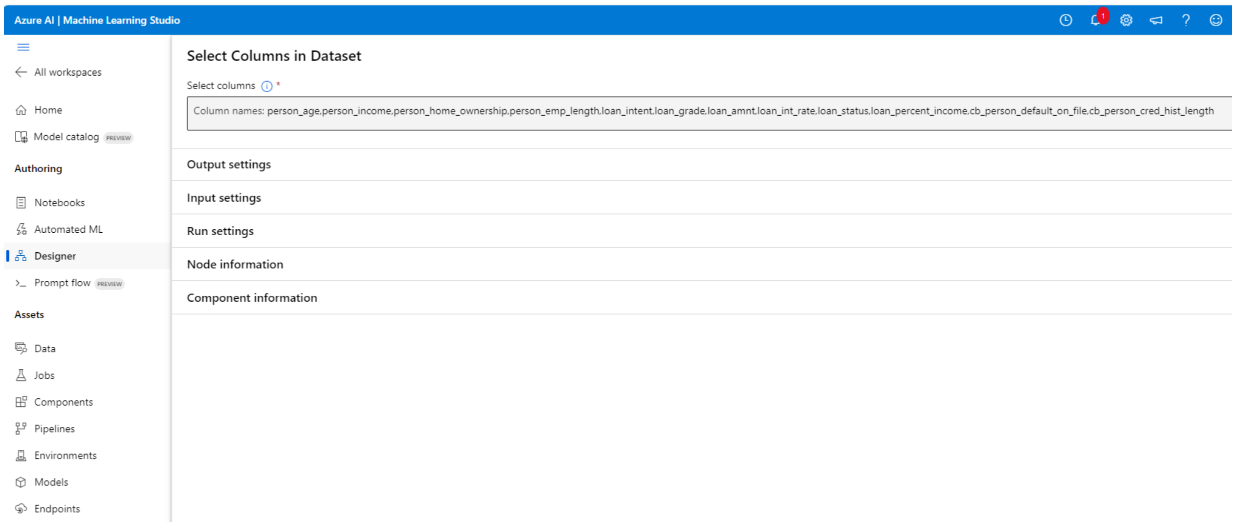


Figure 79 Visualizing the balanced dataset after applying smote component.

5.3.5 Select columns.

The "Select Columns" component in Azure is essential for managing dataset columns in machine learning. It allows certain selection of specific columns to include or exclude in the machine learning pipeline, ensuring that only relevant attributes are retained. This component enables the creation of a customized subset of columns from the original dataset, facilitating efficient data optimization. Users can choose columns based on various criteria, such as "person_income," "person_home_ownership," and others. "Select Columns" streamlines the dataset, enhancing the efficiency of machine learning operations by reducing computational complexity and improving model accuracy. It serves as a valuable data reduction technique and ensures that models are trained and evaluated with the most pertinent attributes.



Select Columns in Dataset

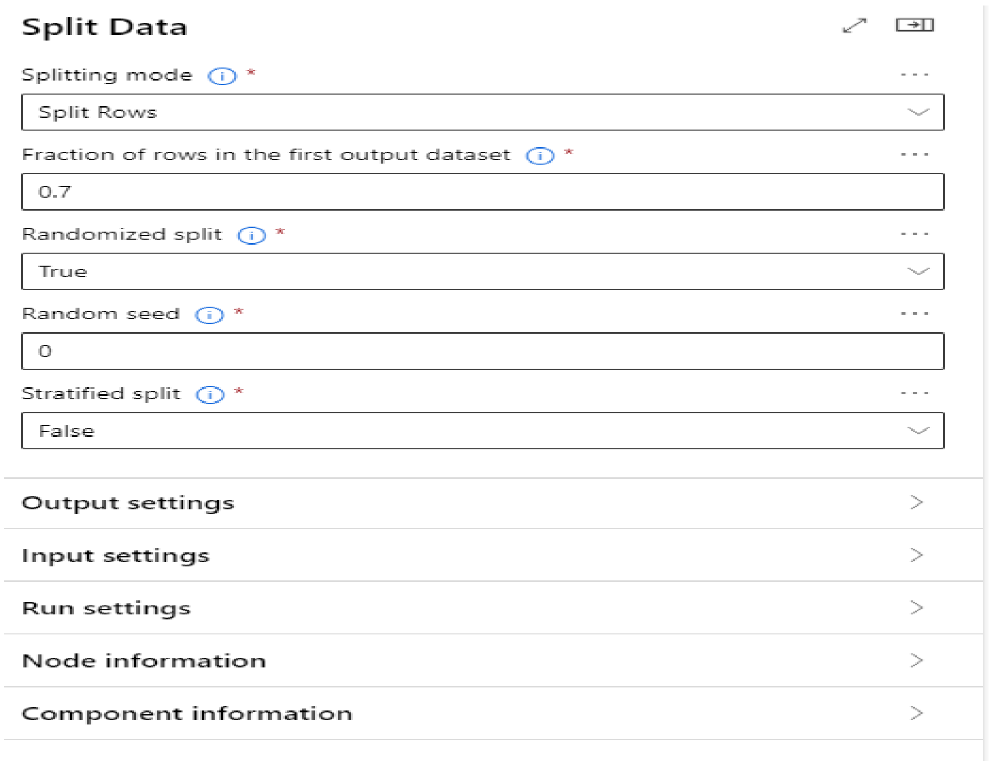
Select columns *

Column names: person_age,person_income,person_home_ownership,person_emp_length,loan_intent,loan_grade,loan_amnt,loan_int_rate,loan_status,loan_percent_income,cb_person_default_on_file,cb_person_cred_hist_length

Figure 80 Select Columns Component in Azure ML

5.3.6 Split Data

The split data is a component in Azure used to divide the dataset into two distinct sets, namely the training set and the testing set. The split data component was added to the pipeline in the designer under the data transformation category. By default, azure split the dataset 50/50, hence there is a need to tune the splitting ratio to better train the model before the application of the machine learning algorithm. The dataset is connected to the splitting component where the rows are divided into the fraction of rows where a value between 0 and 1 is entered which shows the percentage ratio. In this research the ratio between the training and testing set was divided into 0.7 and 0.3 respectively. The roles were split and the output from the split model was trained and scored.


Split Data

Splitting mode *
Split Rows

Fraction of rows in the first output dataset *
0.7

Randomized split *
True

Random seed *
0

Stratified split *
False

Figure 81 Split component in Azure ML studio

5.3.7 Train Model

This component was used to train the classification model. After the model parameters has been set and the required target data has been specified, the train model was selected in the ML component. The label class which is the loan_status was selected as the variable to be trained for the classification technique. The result from the split data was connected to the right of the trained model and on the left side, the untrained model was connected to the ML algorithm to be trained. The result of the trained model was score and evaluated using the score and evaluate component.

5.3.8 Azure Machine learning Algorithms

Machine learning algorithms are sets of instructions which allow computers to make predictions from inputted dataset. In this research Azure machine learning was used for a classification task. These algorithms are used for supervised learning tasks where the goal is to categorize data into predefined classes or labels. Three machine learning algorithms were used such as two-class decision forest, two-class support vector machine, two-class neural network.

5.3.8.1 Two-class decision forest

The Two-class Decision Forest in Azure is a robust ensemble learning algorithm primarily designed for classification tasks. Unlike relying on a single model, it leverages the power of multiple related models to enhance accuracy and performance. Decision Forest shares similarities with Random Forest but distinguishes itself by utilizing the entire dataset at randomized starting points. In this algorithm, the goal is to predict a target variable with a maximum of two classes. Setting up a Two-class Decision Forest in Azure involves configuring parameters within the Azure ML pipeline. It uses bagging resampling, randomly sampling the dataset with replacement to create diverse subsets for training each decision tree. A single-parameter trainer mode is chosen, specifying the creation of 10 decision trees. The maximum depth of the decision trees is set at 32, and each leaf node contains a minimum of 1 sample. This configuration ensures that the algorithm's ensemble of decision trees can make accurate predictions while maintaining flexibility and adaptability to various datasets. The Azure Two-class Decision Forest, which offers improved accuracy and robustness through ensemble learning, is a useful tool for binary classification issues in general. It is a strong addition to the machine learning toolset and shines in situations were making precise predictions between two classes is important.

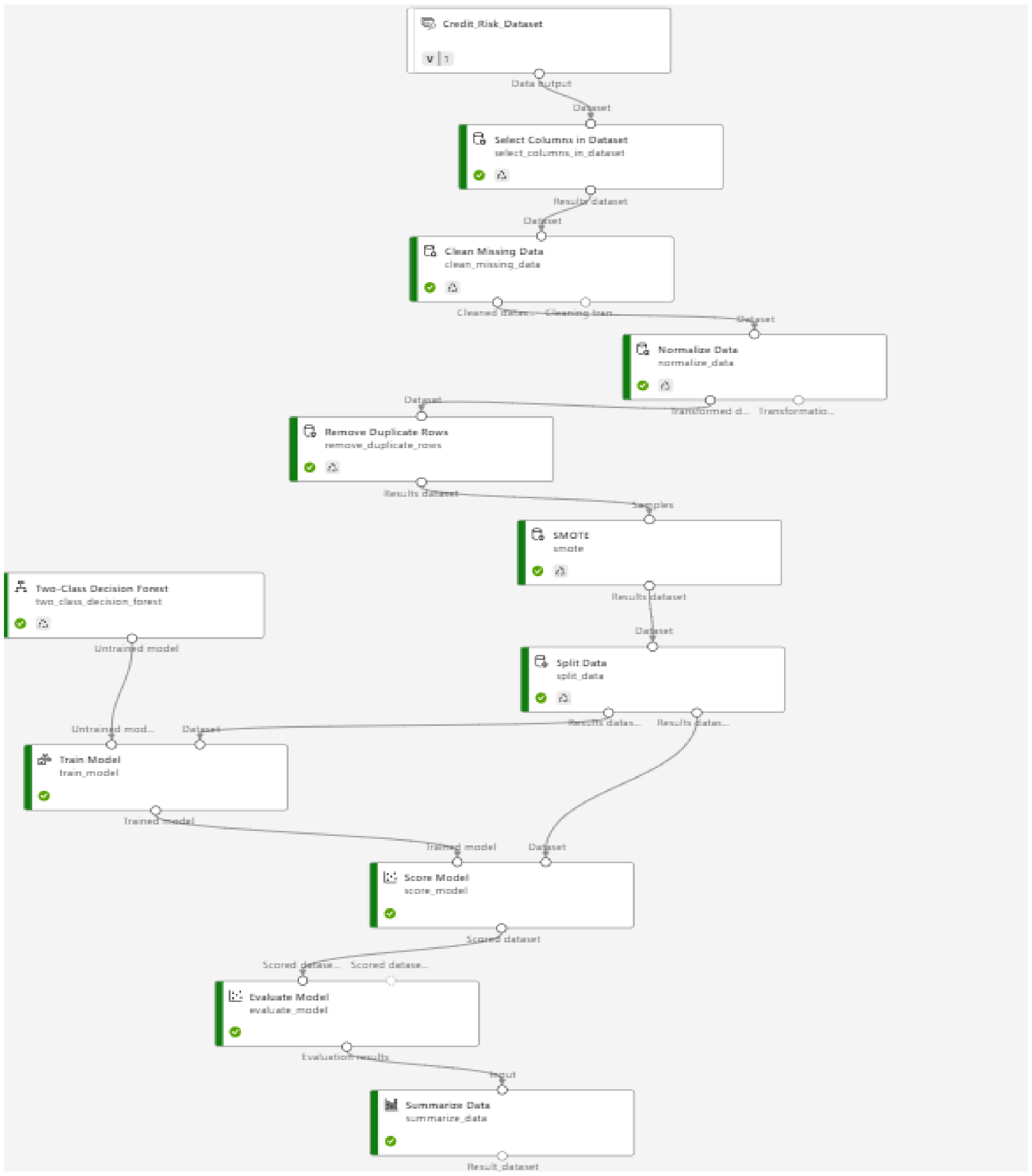


Figure 82 Two-Class Decision Forest ML Model.

5.3.8.2 Two-class Support Vector machine

The Two-Class Support Vector Machine (SVM) in Azure is a powerful supervised machine learning algorithm designed for binary classification tasks, where predictions involve one of two possible classes. SVM excels in recognizing patterns within a hyperplane space, effectively separating input data points for precise classification. In this research, the Two-Class SVM was employed to predict the outcome of a specific target class. Before model training, the dataset underwent essential preprocessing, including normalization to ensure consistent scaling across features. Azure's machine learning pipeline seamlessly integrated the SVM component, making it a pivotal part of the machine learning model. Key configurations for the Two-Class SVM in Azure included setting the number of iterations for model building to 10, allowing for comprehensive training and refinement. Feature normalization was applied to center data points around the mean, promoting uniformity and enhancing overall model performance. To simplify training, the model was configured to a single parameter setting, streamlining the process. Furthermore, the lambda value for weight regularization was set at 0.001 to control overfitting and bolster model generalization. Data normalization was enforced before training to ensure that input data adhered to appropriate standards. The resulting model underwent training and evaluation to ensure its capability to make precise predictions based on input variables. Azure's Two-Class SVM proved to be a potent tool for binary classification tasks, and this research harnessed its capabilities effectively through preprocessing and parameter tuning. The outcome was a robust and highly accurate machine learning model, ready for real-world applications.

$$\boldsymbol{min}\ \frac{\mathbf{1}}{\mathbf{2}} w^{(1)T} w^{(1)} + \gamma \frac{1}{2} \sum_{j=1}^{N^{(1)}} \varepsilon^2$$

Such that $Y_j\left[w^{(1)}\ \varphi\left(X_j\right) + \beta^{(1)}\right] = 1 - \varepsilon_j, j = 1, \dots N^{(1)}$

Where $w^{(1)}$ is the weight vector in primal space and γ is the regularizer.

Equation 16 Support Vector Machine Equation

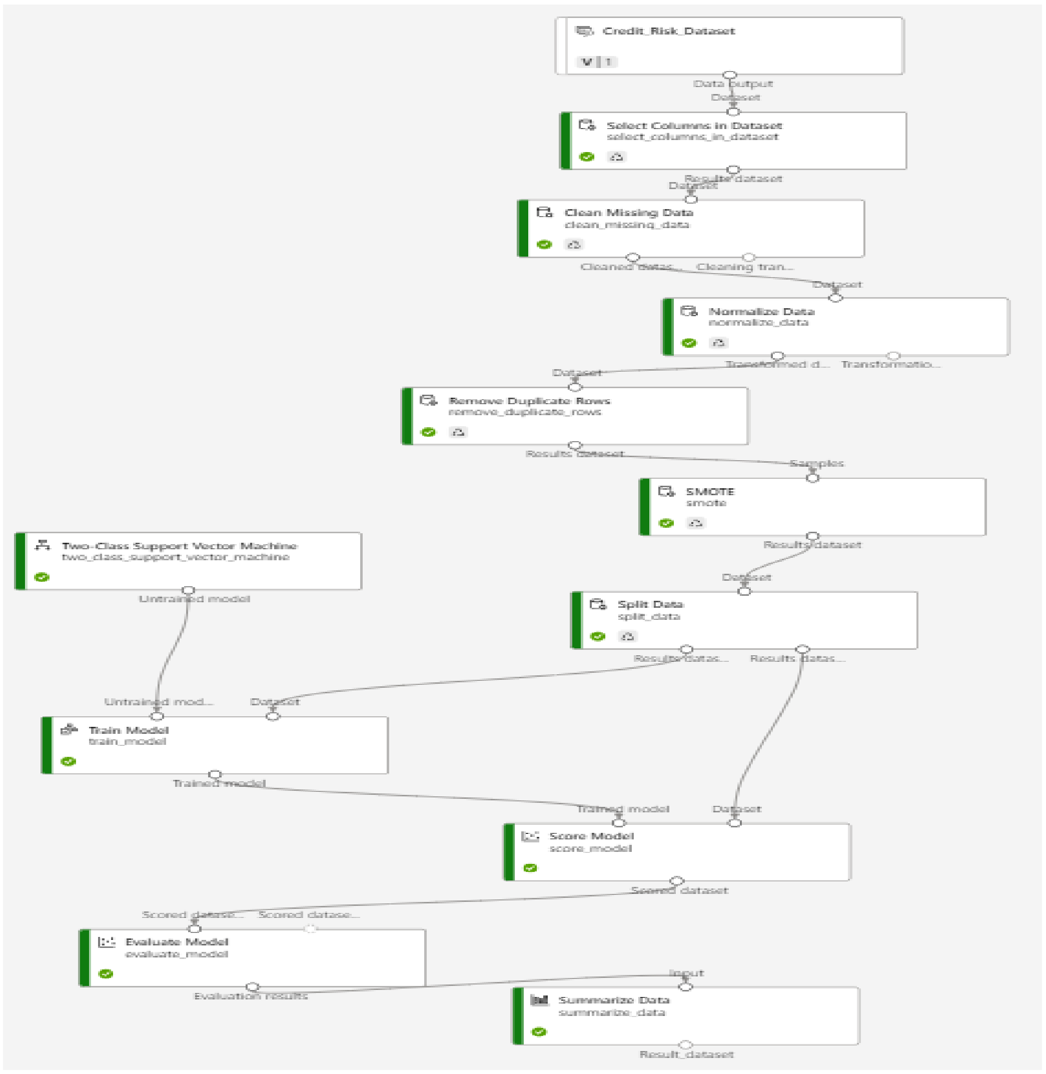


Figure 83 Two-Class SVM Model in Azure ML studio

5.3.8.3 Two-class Neural Network

The Two-Class Neural Network in Azure is a supervised machine learning technique for binary classification. It employs a layered structure consisting of input, hidden, and output layers. These layers are interconnected through weighted edges, forming a weighted acyclic graph. In the network, nodes calculate values by summing weighted inputs and apply activation functions to generate predictions. Training is conducted using labeled data, with the goal of adjusting weights to minimize prediction errors. Azure provides a user-friendly configuration interface for setting up the Two-Class Neural Network. Users can specify parameters like trainer mode, hidden layer specifications (often fully connected), the number of nodes in layers (e.g., 100), and the number of training iterations (e.g., 100). This neural network is a versatile tool for binary classification tasks and is adept at capturing complex patterns in data. Azure's ease of configuration makes it accessible for various applications, from financial predictions to medical diagnosis.. (Khan, 2021)

$$H_j = \varphi\,(\beta_j^{(4)} + \sum_{k=1}^{M} W_{kj}^{(4)}\,X_J$$

Equation 17 Neural Network equation

The output of the classifier is then calculated by combining the hidden nodes using the sigmoid activation function.

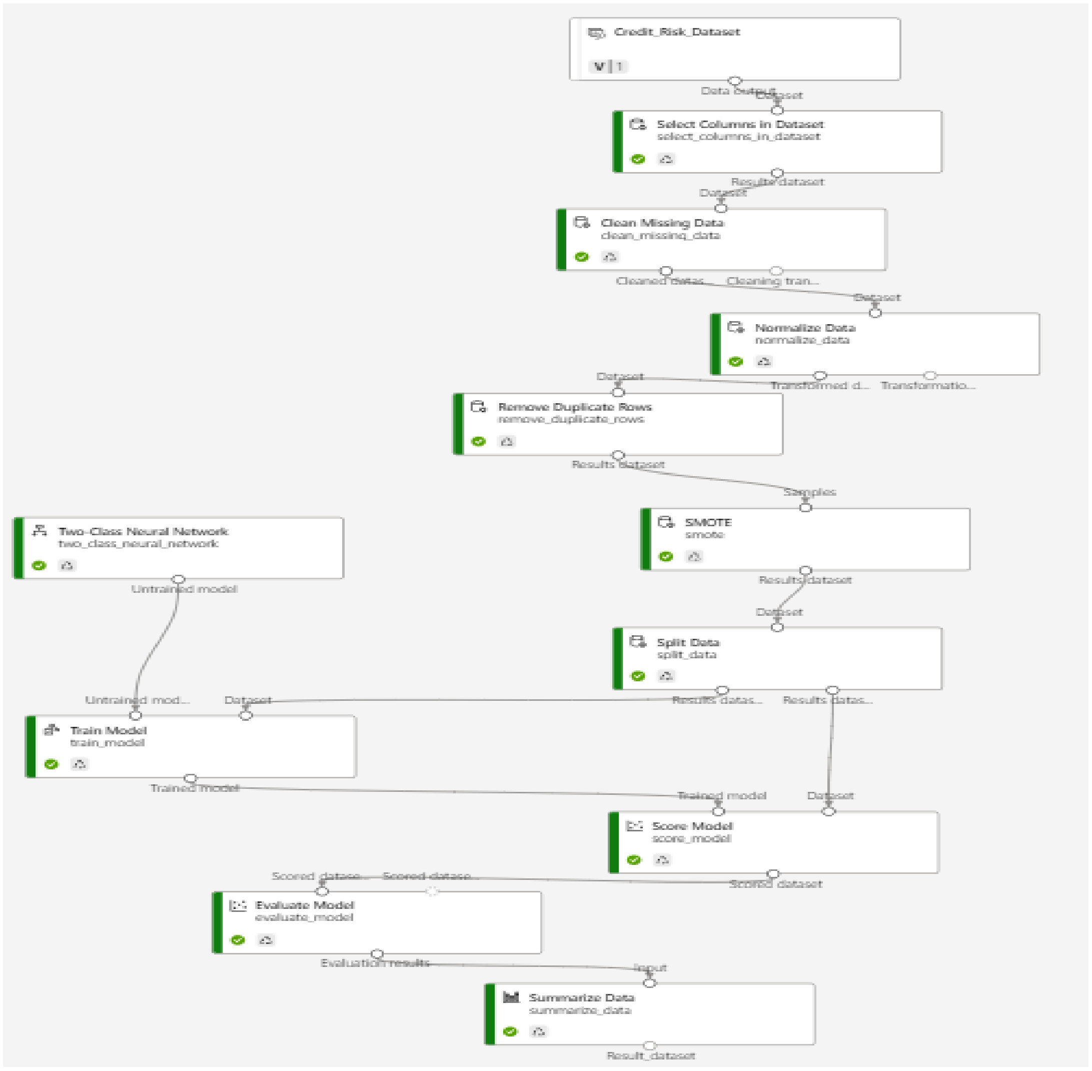


Figure 84 Two-Class Neural Network Model in Azure ML studio

5.3.9 Scored Model

A "scored model" in Azure Machine Learning is the result of a machine learning pipeline that applies a previously trained machine learning model to a new testing dataset. A trained model and a new testing dataset are combined in this procedure. The testing dataset contains data that the model has not encountered in its training data, whereas the model has learnt patterns from the loan data and is registered in the Azure Machine Learning workspace. The model is applied to the testing dataset by the pipeline, which uses the learnt patterns to make predictions or categorize the input data. To evaluate the model's precision and adaptability to new data, the process's outcomes are examined.

5.3.10 Evaluate model.

This component in Azure is vital for assessing a trained machine learning model's accuracy and reliability, particularly in classification tasks. It takes the model's predictions and applies various performance metrics like confusion matrices, accuracy, precision, recall, F1 Score, and AUC to measure its performance. This evaluation process helps determine how well the model classifies data and ensures its effectiveness in making accurate predictions during training.

5.3.10.1 Confusion Matrix

A confusion matrix in Azure evaluates a machine learning model's performance during training. As no model is flawless, some data points are inevitably misclassified. This matrix organizes classifications into a two-by-two structure, with one dimension representing actual values and the other representing predicted values. It counts "true positives" (correct positive predictions), "false positives" (incorrect positive predictions), "false negatives" (incorrect negative predictions), and "true negatives" (correct negative predictions). In this research, the confusion matrix was used to visualize where the model made errors. Predictions are labeled as either positive (P) for correct positive observations or negative (N) for correctly identified non-positives.

5.3.10.2 True Positive (TP) This represents cases where the model correctly predicts a positive outcome (e.g., customer default) based on the observed data. It signifies that the model's prediction aligns with the actual positive class, demonstrating its accuracy in identifying positive instances. In credit risk assessment, TP indicates correctly identified high-risk customers.

5.3.10.3 True Negative (TN) These instances correspond to the model accurately predicting negative outcomes (e.g., no customer default) when the observed data also indicates negative cases. It represents correct predictions of the negative class, often associated with low-risk customers in credit risk assessment.

5.3.10.4 False Positive (FP) This occurs when the model incorrectly predicts a positive outcome despite the actual data suggesting a negative outcome. FP, also known as Type 1 error, implies that the model wrongly identifies cases as positive when they should be negative, potentially leading to unnecessary actions like rejecting creditworthy customers.

5.3.10.5 False Negative (FN) FN arises when the model incorrectly predicts a negative outcome, even though the data indicates a positive outcome. FN, also called Type 2 error, signifies instances where the model fails to identify actual positive cases, possibly resulting in the approval of high-risk customers.

5.3.11 Summarize data.

Summarize data in the Azure Machine Learning designer was used to create statistical summary of each column in the input table. It shows the characteristics of the complete dataset. Values such as the mean, median Standard deviation, mode, range, quantiles etc. are being shown by the summarized data in the statistical function component.

5.4 Results

In the model used for machine learning classification, Azure cloud computing was used while applying some filters for training and evaluation to achieve a better prediction. The result of the proposed framework provides better classification. This model was implemented using predictive classifier on Microsoft Azure ML studio. Splitting, training, and testing of the model for good convergence while selecting the best model for the dataset by applying different ML algorithms. Different performance was used to measure the model performance.

5.4.1 Performance Index

5.4.1.1 Accuracy

Accuracy is an important metric for evaluating model performance. It measures the ratio of correctly predicted observations to the total observations. In this research the correct predictions are composed of true positives and true negatives.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Equation 18 Accuracy formular

5.4.1.2 Precision

This measures the amount of positive divided by the total number of positive predictions. It is the ratio of positive observations to the total predicted positive observations. There is an inherent tradeoff between precision and recall. The higher the precision, the lower the recall and vice versa.

$$Precision = \frac{TP}{TP + FP}$$

Equation 19 Precision Formular

5.4.1.3 Recall

Recall is also known as True positive rate (TPR) it is the ratio of the correctly predicted positive observations to all the observations in the actual class.

$$Recall = \frac{TP}{TP + FN}$$

Equation 20 Recall Formular

5.4.1.4 Lift

This is a metric to assess the performance of classification algorithms. It is the number of true positives against the positive rate. In this research the more the number of true positives the higher the positive rate. The formular for lift is illustrated below.

$$\text{Lift} = \frac{Predicted\ Rate}{Average\ Rate}$$

Equation 21 Lift Formular

5.4.1.5 F-Measure

This measures the harmonic mean of recall and precision. The result of the harmonic mean gives similar values for precision and recall. It is a more comprehensive evaluation metric in comparison to other metrics since it measures the harmonic mean of two metrics simultaneously.

$$F1\ Score = \frac{(2 * Recall * Precision)}{(Recall + Precision)}$$

Equation 22 F1-Score Formular

5.4.1.6 AUC

This performance metric measures the overall performance of a binary classification model. It stands for Area under curve.

5.4.1.7 ROC

ROC curve stands for Receiver Operating Characteristics and shows the graphical representation of the effectiveness of a binary classification. When evaluating binary classification models, the Receiver Operating Characteristic (ROC) curve is an essential tool. It plots True Positive Rate (TPR) against False Positive Rate (FPR) at various classification levels to show how well a model can distinguish between positive and negative classifications. While a curve at the diagonal line reflects random guessing, a steep curve ascending to the top-left corner indicates excellent performance. The performance of the entire model is summarized by the Area Under the Curve (AUC), with higher AUC values denoting stronger discrimination abilities.

5.4.2 Two-Class Support Vector Machine

The implementation of this supervised machine learning model for predicting two possible outcomes was done using Azure ML Studio using a designer component. After defining the model parameter, training the model was done using training component and providing tagged dataset that includes the label as the outcome. The result shows the performance of the model 5 indexes. Table 47.0 shows the confusion matrix for SVM where the number of TP is 6742 and TN is 6051.

ML Algorithms	Accuracy	Precision	Recall	AUC	F1 Score
Two-class SVM	79.40	81.11	79.40	86.90	80.30

Table 47 Performance Index for Two-class SVM

5.4.2.1 ROC Curve for two class support vector machines

ROC represents the receiver operating characteristic (ROC) curve, and it shows the relationship between true positive rate (TPR) and false positive rate (FPR) as the threshold of decision changes. It is most effective for training imbalance datasets. The result shows that at higher threshold the rate of false positives increases.

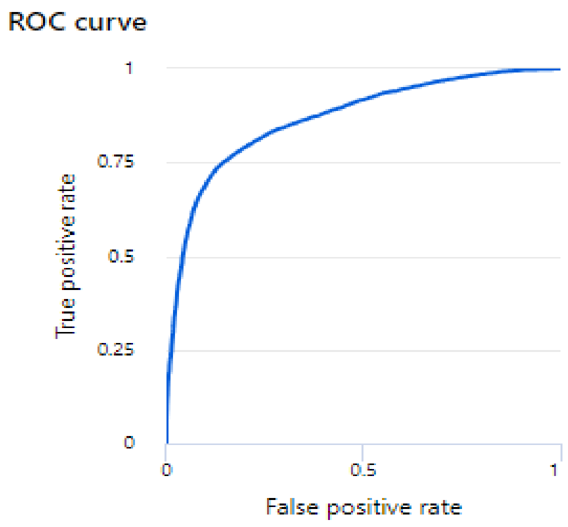


Figure 85 ROC curve for Two-class SVM

5.4.2.2 Precision-recall curve for two-class SVM

It shows the relationship between precision and recall as the decision threshold changes. Precision is the ability of a model to label correctly as positive, on the other hand recall detects all positive samples. The graph in figure 49.0 below shows the lower the precision, the higher the recall and vice-versa.

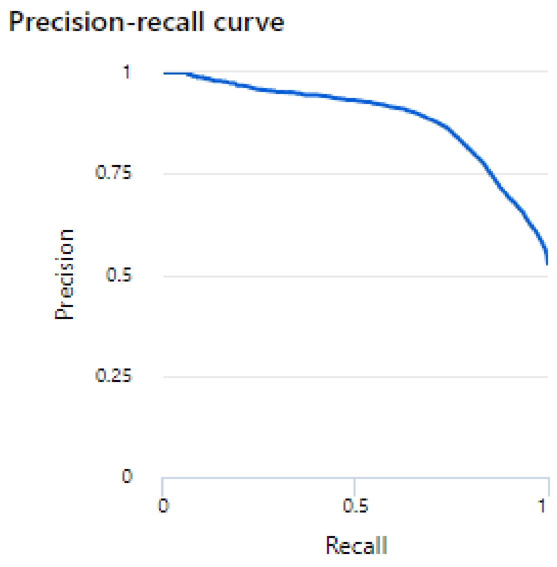


Figure 86 Precision-Recall curve for Two-class SVM

5.4.2.3 Lift curve for two-class SVM

A Lift Curve for a Two-class Support Vector Machine (SVM) evaluates the model's ability to identify positive cases at different threshold values compared to random chance. It plots the lift value (how much better the model is than random) against the positive rate (proportion of positive cases). In the figure 50 below the lift curve shows a linear curve for number of true positives up between 0 - 6,742, there was a change in slope beyond that threshold showing an increase in positive rate.

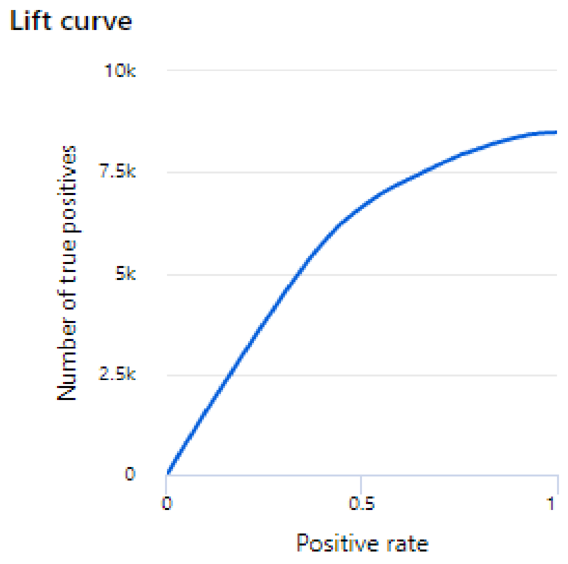


Figure 87 Lift curve for Two-class SVM

5.4.2.4 Confusion Matrix for Two-Class Support Vector Machine

The confusion matrix below shows the actual and predicted class correctly predicted that 6,742 customers are considered non-defaulters, and 6,051 customers are considered defaulters.

Confusion Matrix	Actual 1.0	Actual 0.0
Predicted 1.0	6742	1568
Predicted 0.0	1744	6051

Table 48 Confusion Matrix for Two-class Support Vector Machine

The confusion matrix for the Two-class Support Vector Machine (SVM) consists of True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) with the values illustrated below. In a credit risk assessment scenario, TP represents correctly identified high-risk customers, FP represents falsely classified high-risk customers, FN represents missed high-risk customers, and TN represents correctly identified low-risk customers. Performance metrics include accuracy (81.74%), precision (81.10%), recall (79.45%), specificity (79.40%), and F1-Score (80.77%), which collectively assess the SVM's binary classification performance for credit risk assessment.

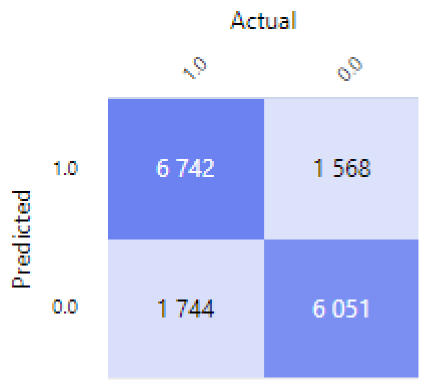


Figure 88 Confusion Matrix for Two-class Support Vector Machine

5.4.3 Two-Class Decision Forest

The table .0 below provides a performance summary of Two-class Decision Forest on a binary classification. The algorithm achieved impressive results with an accuracy of 93.60%, a precision of 94.10%, and a recall of 93.70%, demonstrating its ability to accurately identify positive cases (e.g., loan defaults). Additionally, it attained a high AUC value of 98.10%, indicating strong discrimination capabilities, and a balanced F1 Score of 93.91, showcasing its overall effectiveness in classification. These metrics collectively highlight the algorithm's suitability for credit risk assessment.

ML Algorithms	Accuracy	Precision	Recall	AUC	F1 Score
Two-class Decision Forest	93.60	94.10	93.70	98.10	93.91

Table 49 Classification Report for Two-class Decision Forest

5.4.3.1 ROC curve for Two-class decision forest

The fig 28 below illustrates the ROC curve for a Two-class Decision Forest illustrates the model's ability to differentiate between positive and negative cases at different classification thresholds. The curve's shape and the AUC value provide insights into the model's classification performance. The graph shows a increase in false positive rate at a true positive rate of 93.60.

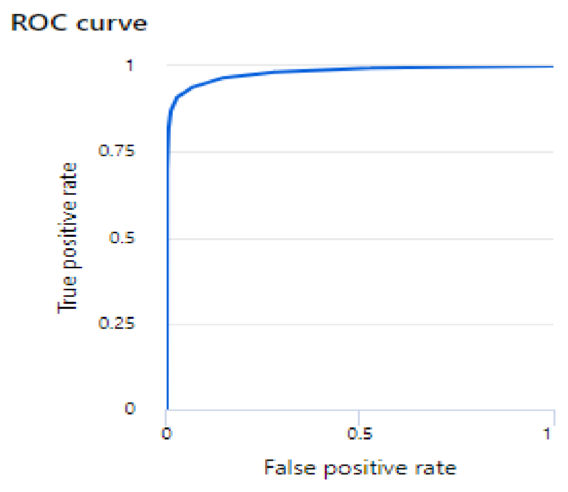


Figure 89 ROC for Two-class Decision Forest

5.4.3.2 Precision-recall curve for Two-class decision forest

The precision-recall curve for the Decision Forest algorithm assesses the model's binary classification performance across different threshold values. It illustrates the trade-off between precision (the accuracy of positive predictions) and recall (the model's ability to identify all positive instances correctly). The curve displays a descending trend, indicating that as the classification threshold decreases, more instances are classified as positive, affecting both precision and recall. A perfect model would have precision and recall both equal to 1, signifying accurate positive predictions.

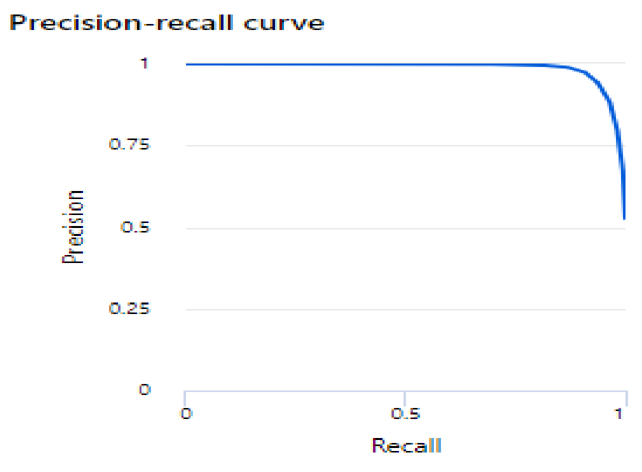


Figure 90 Precision-Recall Curve for Two-Class Decision Forest

5.4.3.3 Lift curve for Two-class decision forest

The lift curve for the Two-class Decision Forest algorithm indicates that it performs well in terms of identifying positive cases. The curve's behavior, particularly the peak at 7,952 true positives, suggests that the model maintains a consistent and high level of accuracy in predicting positive instances as the positive rate increases.

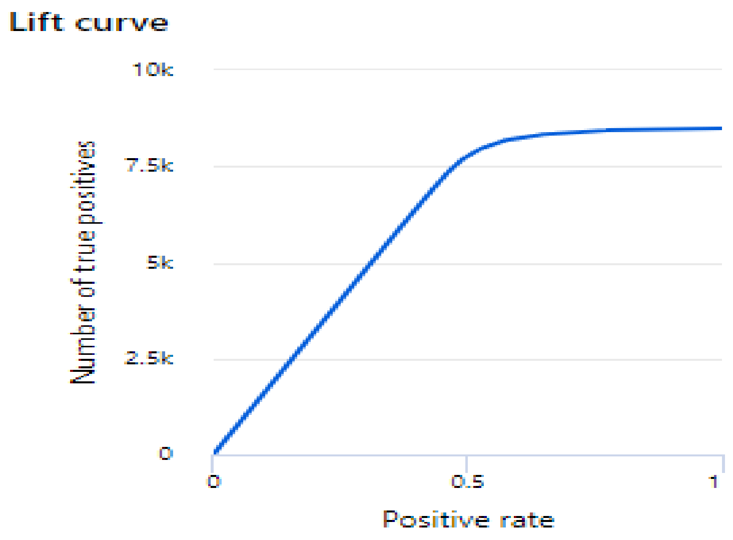


Figure 91 Lift Curve for Two Class Decision Forest

5.4.3.4 Confusion matrix for Two class decision forest

The fig 31.0 below concludes that the model's predictions suggest that the model predicted 7,952 customers would default (True Positives) while correctly identifying 7,119 customers as non-defaulters (True Negatives). While misclassifying the FP and FN to be 500 and 534. This confusion matrix demonstrates the model's ability to make accurate predictions for both defaulting and non-defaulting customers in a credit risk management scenario.

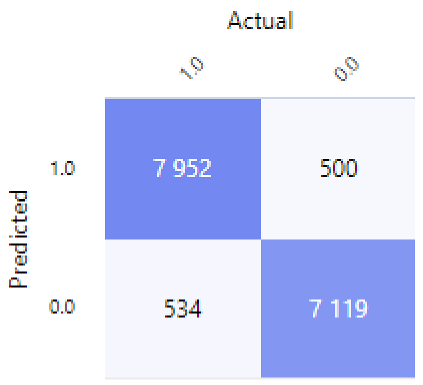


Figure 92 confusion matrix for Two class decision forest

5.4.4 Two-class Neural Network

The classification report pertains to a Two-class Neural Network's performance in a binary classification task. The model achieved an accuracy of 83.0%, indicating its ability to make correct predictions in 83.0% of instances. Precision, which measures the accuracy of positive predictions, stood at 85.40%, showcasing the model's proficiency in avoiding false positives. The recall rate, measuring the model's ability to correctly identify positive instances, was 81.60%. The model's AUC, a metric reflecting its capability to discriminate between positive and negative cases, was 91.80%, signifying good discriminative power. The F1 Score, balancing precision, and recall, reached 83.50%, indicating a well-rounded performance.

ML Algorithms	Accuracy	Precision	Recall	AUC	F1 Score
Two-class Neural Network	83.0	85.40	81.60	91.80	83.50

Table 50 Classification Report for two-class Neural Network

5.4.4.1 ROC curve for Two-Class Neural Network

provides insights into its classification performance at different thresholds, emphasizing the balance between correctly identifying positive cases and accepting false positives. At the true positive rate of 57% there was a steady increase in false positive rate as shown in the graph below.

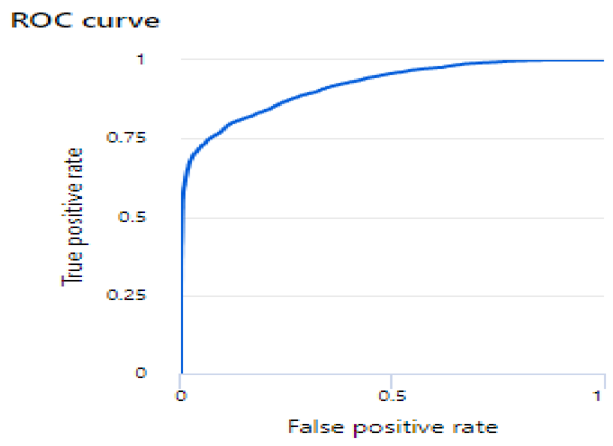


Figure 93 ROC for Two-Class Neural Network

5.4.4.2 Precision -recall curve for Neural Network

In the figure below shows a drop in precision as the recall increases. This drop signifies that when the model uses a threshold of certain value to make binary classification decisions, it becomes less precise, meaning there are more false positive predictions. The closer the recall (the proportion of actual positive cases that the model correctly identifies) gets to 1 (perfect recall), the lower the precision becomes.

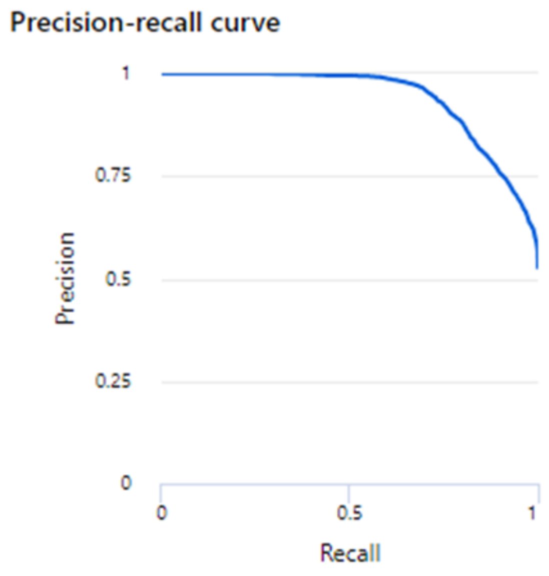


Figure 94 Precision-Recall Curve for Two-Class Neural Network

5.4.4.3 Lift curve for Neural Network

The lift curve in machine learning, as shown below, highlights that as the number of true positive predictions made by a neural network algorithm increase, there is a corresponding increase in the positive rate. This demonstrates the model's ability to make more accurate positive predictions, making it a valuable tool for assessing and visualizing the algorithm's performance, particularly in tasks such as binary classification.

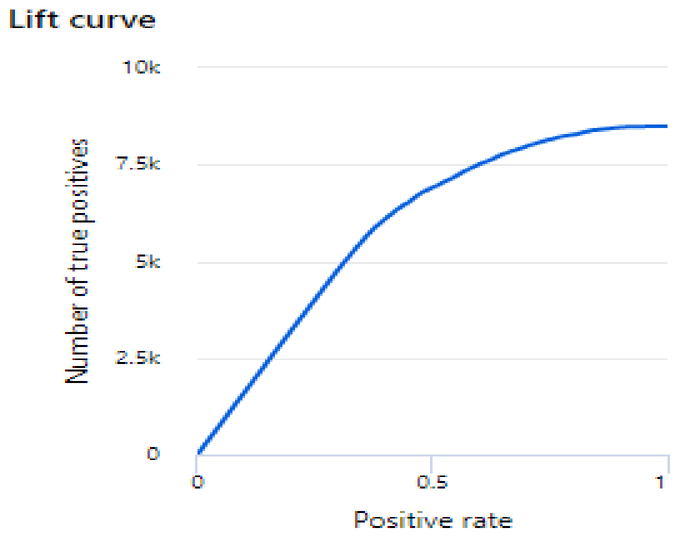


Figure 95 Lift Curve for Two-Class Neural Network

5.4.4.4 Confusion matrix for Neural Network

In figure 35 below reveals that 6,927 customers were correctly classified as true defaults, indicating the algorithm's accuracy in identifying high-risk borrowers. Additionally, 6,434 customers were correctly classified as true non-defaulters, demonstrating the model's ability to accurately identify safe borrowers. The matrix also misclassified the FP and FN to be 1,185 and 1,559 respectively. The confusion matrix serves as a valuable tool for assessing the algorithm's performance in predicting loan defaults and non-defaults in credit risk management.

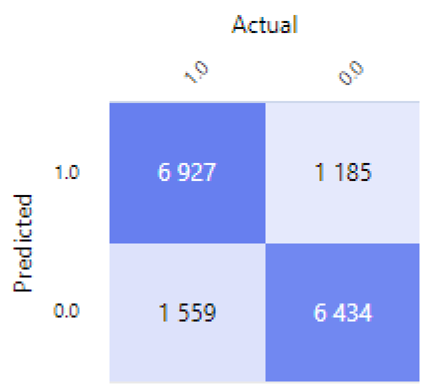


Figure 96 Confusion Matrix for Two-Class Neural Network

5.4.5 Discussion

The summary table provided in figure table 5.0 offers a comprehensive comparison of different machine learning algorithms applied to a binary classification task for loan prediction within the Azure Machine Learning Studio. This analysis was conducted in a supervised learning environment, where the models were trained and evaluated using certain performance metrics. The "scored and evaluate" component of Azure Machine Learning studio presented the outcomes of this analysis, showcasing the varying values of several performance indexes for each of the machine learning algorithms employed. These indexes are crucial for assessing how well the models perform in making binary classification predictions. In this case, five performance indexes were used to gauge the models' predictive capabilities. Among the three machine learning algorithms considered below, the Two-class Decision Forest stands out as the best performer. It achieved an impressive accuracy rate of 93.60%, indicating that 93.60% of the predictions made by this model were correct. Additionally, it demonstrated a high precision score of 94.10%, implying that when it predicted a positive outcome, it was correct 94.10% of the time. The recall score of 93.70% suggests that it successfully identified 93.70% of all actual positive instances. The AUC (Area Under the Receiver Operating Characteristic Curve) value of 98.10% signifies excellent discrimination ability, which is essential in binary classification tasks. Lastly, the F1-Score of 93.91% is a balanced measure that considers both precision and recall and provides a single metric to assess model performance. In summary, the Two-class Decision Forest outperformed the other models in all of these key performance metrics.

5.4.5.1 Summary Table

ML Algorithms	Accuracy	Precision	Recall	AUC	F1 Score
Two-class SVM	79.40	81.11	79.40	86.90	80.30
Two-class Neural Network	83.0	85.40	81.60	91.80	83.50
Two-class Decision Forest	93.60	94.10	93.70	98.10	93.91

Table 51 Summary Table for the Azure ML Algorithms

5.4.6 Conclusion

In the realm of credit risk management, securing adequate funding is essential to prevent business stagnation or eventual collapse. Financial institutions are adept at assessing loan applications even in the face of uncertain conditions. This research introduces a highly effective approach to credit risk management within financial institutions, leveraging Microsoft Azure Machine Learning. The model was designed and developed in the Azure Machine Learning Studio, employing a series of crucial components, including data pre-processing, feature selection, training, testing, and evaluation. Three distinct machine learning algorithms were harnessed for training on the dataset. These encompassed ensemble learning, neural networks, and hyperplane classification. Given that the model is designed for binary classification, the study utilized the Two-class Support Vector Machine, Two-class Neural Network, and Two-class Decision Forest algorithms. Evaluating the model's performance is a pivotal aspect of this research, and as such, five key performance indices were considered. These metrics provide valuable insights into how the model performs during training and testing phases. The results clearly demonstrate that the Two-class Decision Forest algorithm outperformed its counterparts, exhibiting exceptional capabilities in generalizing the dataset. With metrics including accuracy (93.6%), precision (94.10%), recall (93.70%), AUC (Area Under the Curve) (98.10%), and F1-Score (93.91%), this algorithm consistently delivered superior results. Moreover, this model was meticulously prepared for deployment using its REST API, enabling seamless integration and consumption within the financial institution's operational framework. This holistic approach to credit risk management, backed by advanced machine learning techniques and Azure's robust infrastructure, offers a promising solution to mitigate financial risks effectively and make informed lending decisions.

Conclusion

In conclusion, these three research projects demonstrate the invaluable role of advanced technology and machine learning in modern financial risk management. The first project leverages big data technologies like Hadoop and Spark to predict loan defaulters accurately, highlighting the potential for enhancing credit risk management by analyzing customer spending habits. The second project introduces a novel spam-ham-phishing dataset for email classification using machine learning and NLP techniques, showcasing the importance of improving fraud prevention in the financial sector. Finally, the third project explores Azure machine learning's application for identifying loan defaulters and provides valuable insights into the effectiveness of various machine learning algorithms in credit risk management. These studies collectively underscore the critical role of data-driven approaches in strengthening financial risk management, mitigating fraud, and promoting innovation in the financial sector.

REFERENCES

Addo, P. M., Guegan, D., & Hassani, B. (2018). Credit risk analysis using machine and deep learning models. Risks, 6(2), 38.

Alshouiliy, K., AlGhamdi, A., & Agrawal, D. P. (2020). AzureML based analysis and prediction loan borrowers creditworthy. 2020 3rd International Conference on Information and Computer Technologies (ICICT),

Ananthu, S., Sethumadhavan, N., & AG, H. N. (2021). Credit Card Fraud Detection using Apache Spark Analysis. 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI),

Armel, A., & Zaidouni, D. (2019). Fraud detection using apache spark. 2019 5th International Conference on Optimization and Applications (ICOA),

Arun, K., Ishan, G., & Sanmeet, K. (2016). Loan approval prediction based on machine learning approach. IOSR J. Comput. Eng, 18(3), 18-21.

Bacanin, N., Zivkovic, M., Stoean, C., Antonijevic, M., Janicijevic, S., Sarac, M., & Strumberger, I. (2022). Application of natural language processing and machine learning boosted with swarm intelligence for spam email filtering. Mathematics, 10(22), 4173.

Bagui, S., Nandi, D., Bagui, S., & White, R. J. (2021). Machine learning and deep learning for phishing email classification using one-hot encoding. Journal of Computer Science, 17, 610-623.

Bahgat, E. M., Rady, S., Gad, W., & Moawad, I. F. (2018). Efficient email classification approach based on semantic methods. Ain Shams Engineering Journal, 9(4), 3259-3269.

Barnes, J. (2015). Azure machine learning. Microsoft Azure Essentials. 1st ed, Microsoft.

Berrada, I. R., Barramou, F. Z., & Alami, O. B. (2022). A review of Artificial Intelligence approach for credit risk assessment. 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP),

Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4). Springer.

Brownlee, J. (2021). Types of Classification Tasks in Machine Learning. Machine Learning Mastery, 4p. Available online at: https://machinelearningmastery. com/types-of-classification-in~ machine-learning (accessed November 25, 2021).

BUENO, L. (2022). Analyzing Credit Default. Kaggle. https://www.kaggle.com/code/juniorbueno/analyzing-credit-default

Chakravarty, A., & Manikandan, V. (2022). An Intelligent Model of Email Spam Classification. 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT),

Chen, T. (2021). Credit Default Risk Prediction of Lenders with Resampling Methods. 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI),

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining,

Coşer, A., Maer-matei, M. M., & Albu, C. (2019). PREDICTIVE MODELS FOR LOAN DEFAULT RISK ASSESSMENT. Economic Computation & Economic Cybernetics Studies & Research, 53(2).

Feng, W., Sun, J., Zhang, L., Cao, C., & Yang, Q. (2016). A support vector machine based naive Bayes algorithm for spam filtering. 2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC),

Fette, I., Sadeh, N., & Tomasic, A. (2007). Learning to detect phishing emails. Proceedings of the 16th international conference on World Wide Web,

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

Hartwig, F., & Dearing, B. E. (1979). Exploratory data analysis. Sage.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2). Springer.

Hindistan, Y. S., Kıyakoğlu, B. Y., Rezaeinazhad, A. M., Korkmaz, H. E., & Dağ, H. (2019). Alternative credit scoring and classification employing machine learning techniques on a big data platform. 2019 4th International Conference on Computer Science and Engineering (UBMK),

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). Springer.

Junnarkar, A., Adhikari, S., Fagania, J., Chimurkar, P., & Karia, D. (2021). E-mail spam classification via machine learning and natural language processing. 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV),

Kachhwaha, U. K., & Shrivastava, A. (2020). Credit Threat Estimation by Machine Learning Techniques Over Cloud Platform. International Research Journal of Engineering & Applied Sciences (IRJEAS), 8(4), 11-16.

Khan, T. (2021). Application of two-class neural network-based classification model to predict the onset of thyroid disease. 2021 11th International conference on cloud computing, data science & engineering (Confluence),

Koç, U., & Sevgili, T. (2020). Consumer loans' first payment default detection: a predictive model. Turkish Journal of Electrical Engineering and Computer Sciences, 28(1), 167-181.

Kotsiantis, S., Athanasopoulou, E., & Pintelas, P. (2006). Logitboost of multinomial Bayesian classifier for text classification. International Review on Computers and Software (IRECOS), 1(3), 243-250.

Kumar, N., & Sonowal, S. (2020). Email spam detection using machine learning algorithms. 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA),

Kumar, V., & Minz, S. (2014). Feature selection: a literature review. SmartCR, 4(3), 211-229.

Li, X., Long, X., Sun, G., Yang, G., & Li, H. (2018). Overdue prediction of bank loans based on LSTM-SVM. 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI),

Magdy, S., Abouelseoud, Y., & Mikhail, M. (2022). Efficient spam and phishing emails filtering based on deep learning. Computer Networks, 206, 108826.

Misheva, B. H., Giudici, P., & Pediroda, V. (2018). Network-based models to improve credit scoring accuracy. 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA),

Motwani, A., Chaurasiya, P., & Bajaj, G. (2018). Predicting credit worthiness of bank customer with machine learning over cloud. International journal of computer sciences and engineering, 6, 7.

Mujtaba, G., Shuib, L., Raj, R. G., Majeed, N., & Al-Garadi, M. A. (2017). Email classification research trends: review and open issues. IEEE Access, 5, 9044-9064.

Nandhini, S., & KS, J. M. (2020). Performance evaluation of machine learning algorithms for email spam detection. 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE),

Nebojsa Bacanin, M. Z., Catalin Stoean, Milos Antonijevic, Stefana Janicijevic, Marko Sarac, Ivana Strumberger.,Stefana Janicijevic, Marko Sarac and Ivana Strumberger. (2023, 8/14/2023). Application of Natural Language Processing and Machine Learning Boosted with Swarm Intelligence for Spam Email Filtering. MDPI. https://www.mdpi.com/2227-7390/10/22/4173

Olatunji, S. O. (2017). Extreme Learning machines and Support Vector Machines models for email spam detection. 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE),

Padimi, V., Venkata, S., & Devarani, D. (2022). Applying Machine Learning Techniques To Maximize The Performance of Loan Default Prediction. Journal of Neutrosophic and Fuzzy Systems (JNFS), 2(2), 44-56.

Pai, A. (2020). What is Tokenization in NLP? Here's All You Need To Know. In: Retrieved from Analytics Vidhya: https://www. analyticsvidhya. com/blog/2020 ….

Paul, A., Mukherjee, D. P., Das, P., Gangopadhyay, A., Chintha, A. R., & Kundu, S. (2018). Improved random forest for classification. IEEE Transactions on Image Processing, 27(8), 4012-4024.

Peng, Z. (2019). Stocks analysis and prediction using big data analytics. 2019 international conference on intelligent transportation, big data & smart City (ICITBS),

Rácz, A., Bajusz, D., & Héberger, K. (2021). Effect of dataset size and train/test split ratios in QSAR/QSPR multiclass classification. Molecules, 26(4), 1111.

Rajagopal, S., Hareesha, K. S., & Kundapur, P. P. (2020). Performance analysis of binary and multiclass models using azure machine learning. International Journal of Electrical & Computer Engineering (2088-8708), 10(1).

Schütze, H., Manning, C. D., & Raghavan, P. (2008). Introduction to information retrieval (Vol. 39). Cambridge University Press Cambridge.

Serengil, S. I., Imece, S., Tosun, U. G., Buyukbas, E. B., & Koroglu, B. (2021). A Comparative Study of Machine Learning Approaches for Non Performing Loan Prediction. 2021 6th International Conference on Computer Science and Engineering (UBMK),

Shaheen, S. K., & ElFakharany, E. (2018). Predictive analytics for loan default in banking sector using machine learning techniques. 2018 28th International Conference on Computer Theory and Applications (ICCTA),

Shih, D.-H., Hsu, H.-L., & Shih, P.-Y. (2019). A study of early warning system in volume burst risk assessment of stock with Big Data platform. 2019 IEEE 4th international conference on cloud computing and big data analysis (ICCCBDA),

Shivanna, A., & Agrawal, D. P. (2020). Prediction of defaulters using machine learning on Azure ML. 2020 11th IEEE annual information technology, electronics and mobile communication conference (IEMCON), Shoumo, S. Z. H., Dhruba, M. I. M., Hossain, S., Ghani, N. H., Arif, H., & Islam, S. (2019). Application of machine learning in credit risk assessment: a prelude to smart banking. TENCON 2019-2019 IEEE Region 10 Conference (TENCON),

Singh, D. (2020). Classification Modeling with AzureML studio. Retrieved 26 September from https://ieeexplore.ieee.org/abstract/document/7363944

Toolan, F., & Carthy, J. (2010). Feature selection for spam and phishing detection. 2010 eCrime Researchers Summit,

Tumuluru, P., Burra, L. R., Loukya, M., Bhavana, S., CSaiBaba, H., & Sunanda, N. (2022). Comparative Analysis of Customer Loan Approval Prediction using Machine Learning Algorithms. 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS),

varshachoudhary. (2023). What is data normalization? . GeeksforGeeks. https://www.geeksforgeeks.org/what-is-data-normalization/

Verma, P., Goyal, A., & Gigras, Y. (2020). Email phishing: Text classification using natural language processing. Computer Science and Information Technologies, 1(1), 1-12.

Wu, Q. (2022). Real-time Predictive Analysis of Loan Risk with Intelligent Monitoring and Machine Learning Technique. 2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS),

Yadav, S., & Thakur, S. (2017). Bank loan analysis using customer usage data: A big data approach using Hadoop. 2017 2nd International Conference on Telecommunication and Networks (TEL-NET),

Zhang, N., & Yuan, Y. (2012). Phishing detection using neural network. CS229 lecture notes, 301.

Zhao, Y. (2021). Big Data Financial Algorithm Technology Based on Machine Learning Technology. 2021 International Conference on Aviation Safety and Information Technology,

Zhen, W., & Wenjuan, S. (2016). Commercial bank credit risk assessment method based on improved svm. 2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS),

Zhou, H., Sun, G., Fu, S., Liu, J., Zhou, X., & Zhou, J. (2019). A big data mining approach of PSO-based BP neural network for financial risk management with IoT. IEEE Access, 7, 154035-154043.

Zhu, X., Chu, Q., Song, X., Hu, P., & Peng, L. (2023). Explainable prediction of loan default based on machine learning models. Data Science and Management.